

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

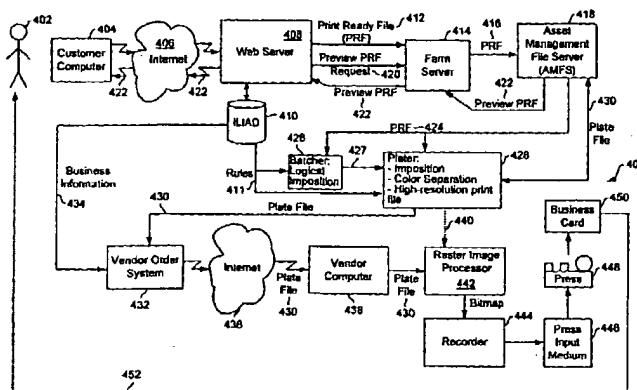




## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification <sup>7</sup> : G06F 17/30, 17/60, G06K 15/00</p>	<p>A1</p>	<p>(11) International Publication Number: WO 00/67153</p>
		<p>(43) International Publication Date: 9 November 2000 (09.11.00)</p>
<p>(21) International Application Number: PCT/US00/10456</p>	<p>(81) Designated States: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).</p>	
<p>(22) International Filing Date: 18 April 2000 (18.04.00)</p>		
<p>(30) Priority Data:</p>		
<p>60/131,716 30 April 1999 (30.04.99) US</p>		
<p>09/460,307 13 December 1999 (13.12.99) US</p>		
<p>(71) Applicant (for all designated States except US): IM-AGEX.COM, INC. [US/US]; Suite 200, 10800 NE 8th Street, Bellevue, WA 98004 (US).</p>		
<p>(72) Inventors; and</p>		
<p>(75) Inventors/Applicants (for US only): KLATT, Cory, E. [US/US]; 14325 - 63rd Avenue West, Edmonds, WA 98026 (US). KRUM, Brent, A. [US/US]; 15908 NE 107th Way, Redmond, WA 98052 (US).</p>		
<p>(74) Agent: WHEELER, Jeffrey, D.; Beyer Weaver &amp; Thomas, LLP, Post Office Box 130, Mountain View, CA 94042-0130 (US).</p>	<p><b>Published</b> <i>With international search report.</i> <i>Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i></p>	

(54) Title: SYSTEM AND FILE STRUCTURE FOR SUPPLYING TO AN INTERNET CUSTOMER BOTH A PREVIEW AND A FINAL PRINT FROM THE SAME PRINT SPECIFICATION FILE



**(57) Abstract**

A system and file structure that produces fast and consistent visual medium materials. Typically, the visual medium will be a printed object, i.e. a business card or the like, resulting from a file which contains text elements, line elements, and graphical elements. A single file is compiled whereby all operations on the file can be completed via reference to the information contained therein. A state flag is associated with each element, the flag having states such as preview, print, both, or none. A global flag is also used having settings such as preview and print. The individual elements within the file can be imaged or printed according to their individual state flag settings as compared to the global flag settings. Files containing graphical elements might embed a graphical element, or provide a link to an element external to the file depending upon the operation desired (e.g. print or preview). For a preview operation, the graphical element is embedded in the file. For a print operation, a link to an externally stored graphical element is provided. For both, a low resolution graphic is embedded within the file and a link to a high resolution graphic is provided. Such links are achieved using Open Prepress Interface comments (or code) or the like around the form of the graphical elements in the files.



*FOR THE PURPOSES OF INFORMATION ONLY*

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		



SYSTEM AND FILE STRUCTURE FOR SUPPLYING TO AN INTERNET CUSTOMER BOTH  
A PREVIEW AND A FINAL PRINT FROM THE SAME PRINT SPECIFICATION FILE

RELATED APPLICATIONS

5        This application claims priority to, under 35 USC 119(e), and also hereby incorporates by reference in its entirety, the Provisional application entitled "A Consistent PostScript System," which was filed on April 30, 1999, by (at least one of) the same inventors in the US Patent and Trademark Office, and assigned application no. 60/131,716.

10       The application also claims priority under 35 USC 119(e), to the Provisional application entitled "Color Washing of Graphic Image Files," which was filed on September 3, 1999, by (at least one of) the same inventors in the US patent and Trademark Office, and assigned application no. 60/152,521.

BACKGROUND OF THE INVENTION

15    1.     Field of the Invention

      The present invention relates to a system and file structure for producing fast and consistent visual medium materials. Typically, the visual medium will be a printed object, i.e. a business card or the like, resulting from a file which contains text elements, line elements, and graphical elements. The elements are arranged and processed in a manner  
20    unique to the present invention. The visual medium might also include an electronic display, such as a computer monitor, or the like.

2.     Description of the Prior Art

      The existing methods of procuring printed business materials are characterized by cumbersome and labor-intensive procedures. These procedures carry with them certain  
25    inefficiencies and are often prone to error. For the majority of small to medium sized printers, the printing of business cards and stationery entails, at a minimum, a time-consuming series of steps, which generally must be repeated every time a new order is placed.

      Referring to Figure 1, a representative block diagram 100 is shown of certain steps  
30    that might be used to procure an order by a customer. While the order might consist of any textual or graphical material, a business card example is used throughout to facilitate



discussion. In general, an administrator in an organization must collect data from the employee who desires the business cards. Such data might include name, title, telephone and fax numbers, mobile phone number, e-mail address, etc. This data then gets telephoned or faxed to the printer as shown in step 102. The printer typesets the information in step 104, and then faxes back a proof to the administrator in step 106. The administrator then typically sends the proof to the employee for verification, and receives the returned proof with marked-up corrections. The proof is then sent back to the printer in step 108. The printer typesets the corrections in step 110 and faxes the proof back to the customer in step 112. Steps 108-112 might be repeated several times until the customer approves the proof in step 114. After the order is final, the printer must go through several additional steps before the order is printed. For instance, the customer service department might enter the job into the printer's internal tracking and billing system. The job then goes to the Prepress department in step 116, which reproduces the content into a format so that it is ready to print. The manufacturing process is applied in step 118 and the order is printed.

Getting through this queue of events with the printer usually takes several days. Seven to ten days after this process is completed, the cards are received by the employee who ordered them. Because each job is entered manually, a new order for a similar customer may not look precisely like the last one. Add the complexities of a multi-location organization (with many employees) and a relatively simple product purchase can become very complex.

Moreover, the printing of more complex items, such as full color pamphlets and brochures and the like, results in many more iterations, typically between the design agency, the customer, and the Prepress department or provider. The iterations due to color correction and perfection of all design elements will likely result in even more time towards finalizing the product. Despite the iterative steps described above, it is estimated that 15 to 30% of print jobs for traditional business materials arrive at the customer with errors.

The propensity for errors, and the general lack of consistency produced using the traditional process, is due, in large part, to the manual nature of the task (for instance, the pre-process step is very manual). At each step in the process, the file may be opened and manipulated repeatedly, which introduces new opportunities for errors and inconsistencies. Figure 2 illustrates a prior art block diagram 200 of representative steps in the process. The sections below, in association with Figure 2, will describe certain stages in the traditional process and delineate potential problems.



## Preview

The process begins with a customer providing the print vendor with the information to be composed on the product. The customer will typically provide the information on an order form, make annotations to a physical sample, and/or communicate the data verbally.

5 The print vendor's job is to create a layout of the print product for the customer to preview and approve. The print vendor will typically interpret the customer's information and compose a preview layout of the product in a publishing tool such as Pagemaker or Quark XPress. In Figure 2, this is shown by the print vendor computer 202 creating a preview layout 204, which results in a preview layout file 206.

10 Unfortunately, this task is made more complicated by a common practice called "mastering". To control costs in printing, it is common to pre-print or "master" stock in bulk with certain static elements. In many cases the static elements are "spot color" or "process color" graphics (while the variable information is usually in a single color, often black). In order to provide a preview of what the printed product will actually look like, the  
15 preview layout must contain both the variable information and the mastered elements. Once the preview layout is completed, it is then printed and faxed to the customer for their approval.

The customer then reviews the faxed proof, annotates any changes, faxes the proof back to the vendor and/or communicates the changes to the vendor verbally. Once the  
20 customer approves the preview layout, the vendor begins the Prepress process. It is important to note that the "preview" that the customer is approving is a faxed copy of a low-quality print out. Because the quality is so low, it is possible (even under the best of conditions) that the final printed product may look slightly different from the proof that the customer approved. If the customer is very demanding or detail-oriented, these differences  
25 may not be acceptable and will require that the vendor re-print the order.

## Composition

Step 208 in Figure 2 shows the next process step of composition. In particular, now that the customer has approved the item, the vendor must create a layout that is suitable for  
30 printing. To do this, all of the mastered elements that were included in the preview layout must be removed. This means that the vendor must open the preview layout file and manipulate the file data manually (or "by hand"). This is problematic, however, because the



vendor is changing a file (or data structure) that the customer has already approved. It is possible that alterations will be made, either intentionally or accidentally, that will change the content or appearance of the product when it is finally printed. Step 210 shows the print layout file, as so modified.

5           The errors that can occur are numerous and varied. Even simple procedures can result in major problems. One simple example, for the purposes of demonstration, is the use of "keyboard shortcuts". Many professionals use a series of keyboard shortcuts (as offered by various programs) instead of a mouse (or other pointing device) to save time in performing simple tasks. These shortcuts typically require the user to press a modifier key  
10 (such as "ALT" or "CTRL") and then press the desired shortcut key. Sometimes, however, the user will mistype and accidentally end up inserting text into a document inadvertently. For example, if the user is trying to cut a graphic or piece of text from a document, the user might use the keyboard shortcut for "Cut" (which might be CTRL-X). If the user fails to fully depress the CTRL key, the letter "x" may be inserted into the document. While this a  
15 relatively straightforward problem, such mistakes might not be detected until late in the process. This might require the vendor to re-print the product, which is expensive and time-consuming. Hence, any reduction in the overall risk of introducing human intervention into the process would be advantageous.

          Still other common problems involve maintaining consistency in at least the  
20 following: object spacing, lines, margins, color adjustment and selection, font spacing, justifications, kerning, and leading. The goal, as yet unattained by prior devices or methods, is to ensure the integrity of text, line and graphic elements in both print and preview operations.

          As another representative example, the act of opening the file can lead to the  
25 common problem of "font substitution." Note that the preview layout file does not (generally) contain the font data necessary to display the text. To save space, the file simply refers to a font file that is stored on the computer used to open the document. If the computer does not have one or more of the fonts referred to by the preview layout file, the closest possible match will generally be substituted from the fonts available. This is known  
30 as "font substitution." Publishing programs may not inform the user that font substitution is taking place, which means that if the user does not notice the swap, then the substituted fonts will be saved with the new document.



When the vendor finally exports the data as a PostScript file for printing, the file will refer to the substituted fonts, not the original fonts. Sometimes the substituted fonts are very similar to the correct fonts, so they might look fine. However, in most cases the substituted fonts are significantly different, and this can cause the final printed product to look vastly different from the preview. Typical problems range from low impact results (e.g. the text looking slightly different), to severe differences (e.g. the text wrapping onto multiple lines, the text coming out completely garbled, etc.). Because final proofing will not be done until later in the process, these problems are often very costly to fix when (and if) they are eventually found.

### Imposition

Step 212 next shows the imposition being performed on the print layout file 210. Imposition is the process of preparing the "print layout" for production on a press. The main goal of imposition is to arrange multiple pages and/or images in the proper order for efficient printing. For example, it is far more efficient to impose four or more business cards onto a single plate than to print each business card individually. The imposition process also requires the addition of elements such as crop marks, registration marks, fold marks, color bars, die marks, and the like to the original print layout file. Imposition can be performed manually or via an automated program.

### Manual Imposition

Manual imposition is often performed on a different computer than that used to produce the preview layout. In many cases this step is even performed by a different person, introducing more opportunities for errors. To impose a plate, the vendor must open the original print layout file and add one or more additional print layouts to create an Imposed Layout File, shown as 214. (It is important to note that some customers like to approve the final imposed layout. As a result, some vendors perform imposition during the preview stage.) Because the imposition process is manual, the errors common to the composition stage can also occur during imposition. Another problem is that because the traditional process for print production is so time-consuming, the information that is to appear on an order may change during the process. In many cases, additional last-minute



orders can be added by the customer at any stage in the process, requiring the vendor to go back and make changes to the imposed layout.

### **Automated Imposition**

5        Some vendors use software products such as Preps by ScenicSoft to build the imposed layout file. Although automated imposition is less susceptible to human error, the process is less than foolproof. For example, it is common for the automated imposition tool to run on a different computer than the original system used to produce the preview layout. This means that the layout file, when exposed to the automated imposition process is subject  
10        to, for instance, font substitution errors, graphic substitution errors, and the like.

### **Color Separation**

Color separation, as shown in step 216, is the process of separating a color image into a series of single color images that will be used to produce plates. When each single-  
15        color plate is printed on top of one another, the result is a composite color image. The color separation step produces an imposed color separated file 218.

In many cases color separation is performed by a RIP (Raster Image Processor). Sometimes, however, the imposed layout file must be color separated prior to the RIP, which means that the vendor must use another software program. In such cases, errors  
20        including font and graphic substitution, can occur just as they did in the composition and imposition stages.

### **Printing**

Once an imposed color separated file is produced it is converted 220 to PostScript, or a plate file 222, for processing by a RIP 224. There are many techniques used to create  
25        PostScript files. Depending on the workflow employed by the print vendor, the PostScript file may include font subsetting as well as OPI (Open Prepress Interface) comments (or executable segments of code) that are processed by the RIP device. In either of these cases, it is possible to introduce font and graphic substitution errors. The output from the RIP  
30        (which is generally a bitmap file) is sent to an output device 226, which might include a Recorder or Image Setter. The output device 226 places the image on a medium to be used



by the press device 228. Such media might include film or digiplate, with direct to plate referring to a media process for taking the image and putting it on a media. Alternatively, the digital or binary file 230 could be received directly by a digital press device 232 for printing.

5

### Prior Electronic Solutions

Based upon the above described problems with the traditional process, different prior electronic solutions have been proposed. While such solutions have attempted to solve the consistency problem in processing a print job, they have generally proven to be insufficient. Below are detailed certain example solutions, and example problems associated with each solution.

One proposed solution includes attempting to automate the process of preview and print layout generation by writing custom plug-ins to commercially-available publishing programs such as Quark XPress or Adobe PageMaker. Example drawbacks to this solution include: (1) The modified software cannot generally keep track of which elements in a layout are mastered and which are not. This means that at least two PostScript files must be generated: one for the preview layout and one for the print layout; and (2) The modified software lacks the ability to store special production information in the PostScript file, such as ink codes, stock information, and other details. The overall solution therefore relies on humans to properly recall this information, or the solution must retrieve such information from yet another document, without any assurance of the accuracy of the production information in the other document. Moreover, these systems do not maintain a reference for standard corporate design or procurement rules for printed matter, and as such are prone to error and not susceptible to automated validation.

Another solution involves using Open Prepress Interface, or OPI. The "OPI Specification 1.3" from Adobe Corporation (the document herein incorporated by reference) defines the Open Prepress Interface (OPI) as a collection of PostScript-language comment conventions that allows a page-layout program to use low- or medium resolution TIFF images for layout and proofing, and have a Prepress system or OPI server automatically substitute a high resolution TIFF or other image when the final film or plates are generated. Both desktop Prepress software and high-end Prepress systems can use OPI comments to minimize network traffic and image storage requirements.



Certain functionality is desired, however, which OPI does not inherently provide. Example of such drawbacks include, but are not limited to, the following: (1) OPI does not generate preview or print layouts. It simply provides a low-resolution file for display on screen and then provides a high-resolution counterpart that is used when it comes time to  
5 print. (2) OPI itself cannot determine whether the system is printing the preview layout or the print layout. Moreover, even if OPI could discern which layout it is printing, it lacks the logic to decide which image to use in which situation. (3) OPI only works for graphic images. For instance, it cannot be used to control text data.

Still other processes have tried to solve the consistency problem by using a simple  
10 Internet solution. Such solutions involve an on-line web site for product ordering that also generates low-res bitmap of preview images that are displayed to the customer at order time for approval. Certain drawbacks of these solutions include: (1) Their bitmap file formats are unable to differentiate between mastered and non-mastered elements such as graphics or text, so the system must generate one low-res bitmap image for preview and another high-  
15 res image for the other stages of the production process. (2) The bitmap images used in these solutions cannot store production-specific information such as ink codes, stock information, and other details. Hence, such solutions generally reproduce the problems associated with the traditional process, but in a computer-controlled environment.

Still another solution might involve an implementation using some form of  
20 programming language. PostScript (for instance) is a programming language for imaging devices. Many solutions propose using some form of PostScript programming. However, it should be noted that the PostScript language, by itself, does not constitute a complete solution to the aforementioned problems. For example, the PostScript language does not contain any concept that differentiates between preview and print elements and cannot  
25 automatically solve the problems with consistency in the print process.

Accordingly, a solution is needed which will gather customer data and provide a consistent output file for use by a printing system. In general, the system should eliminate the various sources of error described above. The system should differentiate between print and preview jobs in order to speed user interaction. Moreover, the system should allow the  
30 user to freely edit and preview the desired print job in a fast and efficient manner. A single, consistent output file should be produced for use by a printer, or other output or processing destination.



SUMMARY OF THE INVENTION

In response to aforementioned costly, cumbersome and error-prone environment, the present invention utilizes certain technology, along with an interface medium such as the Internet, to offer a fully automated, efficient and cost-effective solution for producing print  
5 jobs and the like. The present invention reduces the number of times that human intervention is required in the process and thereby reduces labor intensity, labor cost, time, and high error rates.

In order to eliminate the numerous opportunities for error that appear in the many stages of the traditional print process, the present invention utilizes a single electronic file  
10 format, or a "Print Ready File" (PRF) format, that can be used at all stages of the process. This format provides the ability to tag certain elements to indicate whether they should be included in the preview layout, the print layout, or both. At each stage, the software programs that read and process the information check these tags to determine the exact content required at that stage.

The Print Ready File has each element precisely mapped. Because no human is  
15 required to alter it, the data for the product and the location of its elements need not change. This eliminates a large source of human error. Additionally, because the present system uses the PostScript language (or its equivalents), the present system does not necessarily need to employ commercial page layout software. The present system allows the font and  
20 graphic information to be embedded directly into the file. This eliminates errors, including for instance the font and graphic substitution errors common to the traditional process.

Thus, the present solution allows the creation of a single file that contains all of the data needed for the entire process. The file that the customer previews is the same file that is sent to the vendor (or ultimately the printer). This leads to consistency throughout the life  
25 of a print order, and provides for extremely low error rates overall.

According to one aspect, an Internet site is built to provide a printing service embodying the present invention. Once the Internet site is set up, the customer can modify, order, proof, and control its printed and graphic materials. The solution provided by the present invention eliminates the back and forth process between the customer and the  
30 printer. Additionally, the present invention bypasses the printer's customer service and Prepress processes, (i.e. human intervention processes) altogether. The sequence of events is (in a generalized manner) as follows: (1) The customer inputs the data on the Internet site



of the present invention. (2) The customer approves the proof on-screen. (3) The document is sent as a single PostScript-language file (or its functional equivalent) directly to the printer. (4) The printing plate is made from the digital file. As a result, the printing process is simpler, faster, and less prone to error, resulting in printed business materials that conform according to the customer's corporate specifications.

According to another aspect, the aforementioned single file -- or Print Ready File -- contains line elements, text elements, and/or graphical elements. Each file is assigned a global flag with states for indicating whether a print or preview operation should be performed on the Print Ready File. Each element is assigned a state flag with states such as print, preview, both (or none) for indicating which operations should be performed on the individual elements within the file. The user is provided the ability to set these state flags for each element. Processor-based comments are used, which are generally processed only during print operations. For the different print and preview operations, the file utilizes a different method for accessing graphical elements. A preview operation embeds the graphical element directly in the file, with a link to a blank graphical file. A print operation embeds a blank graphical element, and provides a link to an external file with the graphical element. A "both" operation embeds a low resolution graphical element in the file for preview operations, and provides a link to a high resolution graphical element for print operations. As such, the Print Ready File can be kept to a more manageable size, and yet still use whatever graphical elements are necessary to complete any particular operation.

Still another aspect of the present invention provides for a database with various rules for completing a customer's order. The rules are developed for a particular user's printing needs and are used to properly define the end Print Ready File result. Various assets pertaining to the customer (i.e. logos and such) are stored in yet another database (or file server) and are retrieved (as needed) to complete the Print Ready File, which is then either previewed, or sent through the batching, plating, and final press processes.

These and other aspects and advantages of the present invention will become apparent upon analysis of the following detailed descriptions and studying the various figures and drawings.



BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a representative prior art block diagram of certain steps found in a traditional print job ordering process.

5 Figure 2 illustrates a representative prior art block diagram of certain steps found in the traditional creation of printable material by a print vendor.

Figure 3 illustrates, according to one aspect of the present invention, a generalized series of steps used in creating a print order.

Figure 4 illustrates, according to one aspect of the present invention, a block diagram of the overall system.

10 Figure 4a illustrates, according to one aspect of the present invention, further details regarding the ILIAD element of Figure 4.

Figure 4b illustrates, according to one aspect of the present invention, further details regarding the IOPC element as incorporated with the ILIAD element of Figure 4a.

15 Figure 4c illustrates, according to one aspect of the present invention, further details regarding the Asset Management File Server of Figure 4.

Figure 5a illustrates, according to one aspect of the present invention, a basic data construct of certain elements of a Print Ready File.

Figure 5b illustrates, according to one aspect of the present invention, a flowchart of the global flag comparison logic referred to in Figure 5a.

20 Figure 5c illustrates, according to one aspect of the present invention, a representative construct of certain elements of a Print Ready File when the "preview" state flag is set on the graphic element shown.

Figure 5d illustrates, according to one aspect of the present invention, a representative construct of certain elements of a Print Ready File when the "print" state flag is set on the graphic element shown.

25 Figure 5e illustrates, according to one aspect of the present invention, a representative construct of certain elements of a Print Ready File when the "both" state flag is set on the graphic element shown.

Figure 6 illustrates, according to one aspect of the present invention, a resulting preview to the user of a sample product, e.g. a business card.

30



Figure 7 illustrates, according to one aspect of the present invention, the representative business card of Figure 6 after the imposition stage.

Figure 8a illustrates, according to one aspect of the present invention, a first representative color separation of the business card of Figure 6.

5        Figure 8b illustrates, according to one aspect of the present invention, a second representative color separation of the business card of Figure 6.

Figure 8c illustrates, according to one aspect of the present invention, a third representative color separation of the business card of Figure 6.

10       Figure 9 illustrates, according to one aspect of the present invention, a representative flowchart of the overall process.

Figure 10 illustrates, according to one aspect of the present invention, further representative steps comprising the "customer setup" element of Figure 9.

Figure 11 illustrates, according to one aspect of the present invention, further representative steps of the "create PRF (composition)" element of Figure 9.

15       Figure 12 illustrates, according to one aspect of the present invention, further representative steps of the "generate preview file" element of Figure 9.

Figure 13 illustrates, according to one aspect of the present invention, further representative steps of the "imposition" element of Figure 9.

20       Figure 14 illustrates, according to one aspect of the present invention, further representative steps of the "color separation" element of Figure 9.

Figures 15A and 15B illustrate a computer system suitable for implementing embodiments of the present invention.

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

25       An invention is described herein for improving the efficiency and consistency in generating a print job from customer data (i.e. textual, graphical, and line element information). In the following description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art, that the present invention may be practiced without some or all of  
30       these specific details. In other instances, well known structures, software, devices, and/or



process steps have not been described in detail, so as to not unnecessarily obscure the present invention.

For ease of discussion, the following detailed description is made with reference to the generation and printing of a business card. It should be kept in mind that the inventive  
5 concepts disclosed herein apply equally well to many other types of materials such as film, screens, overlays, cloth, and printed matter such as letterhead, envelopes, notepads, posters, newsletters, coffee mugs, pens, hats, shirts, and other printable materials, and also electronic materials such as vcards, web pages, email, etc. In accordance with one aspect of the present invention the Adobe PostScript language is used. However, any other functional  
10 equivalent might be used for image generation according to a set of programming language instructions. Similarly, where other product examples are referred to, or used to achieve an end result, the same functional equivalent might also be used within the spirit and scope of the present invention.

The system of the present invention includes two main pieces: the Print Ready File  
15 format, which stores the PostScript data according to the present invention, and the related PostScript applications, which read and process the data in the file format according to the present invention. In addition, an Internet-based ordering system provides the customer with the ability to interact with the system to preview and approve orders. The figures below will provide an overview of the ordering system in order to demonstrate the context  
20 in which customers make use of the system. The detailed description will further provide a detailed description of the Print Ready File format and how it works with the related PostScript applications. It should be noted that the present invention would also work with other ordering techniques. The Internet-based ordering system described below is one example of how the invention may be used.

25

### PRINT READY FILE SYSTEM

Figure 3 shows a block diagram 300 a generalized series of steps used in creating a print order. A customer 302 contacts a website via the computer 304. The customer inputs data on the website according to data prompts needed to generate the customer's desired  
30 print job. The system of the present invention creates a Print Ready File (PRF), as shown in element 306. The PRF 306 is shown to the customer 302 for on-screen proofing 308 of various elements comprising the product. Once the order is approved, step 310 shows the



order being sent to the printer via the engine of the present invention (described in further detail below). The PRF 306 is thereafter sent to printer as a print order 312, and the manufacturing (or printing) process begins. The steps of the present invention provide an elapsed time to order of approximately 10 minutes, or less.

5       As an overview to an Internet-based ordering system, a "setup phase" is used wherein the Print Ready File system is configured to produce a Print Ready File for each of the products that the customer wishes to order. In addition, the Print Ready File system also configures an Internet front-end to provide a custom web site for that customer. The customer goes to a web site and selects a particular product to order. The web site loads a  
10       pre-configured order form for the selected product, and the customer enters the data they wish to appear on the card. The web site then transmits the data to the Print Ready File system, which generates the Print Ready File (e.g. as a unique PostScript file).

      The web site takes this Print Ready File and uses it to create the preview layout. It does this by sending the Print Ready File to a viewer program (i.e. the Adobe Acrobat  
15       Distiller program), which reads the Print Ready File and creates a Portable Document Format (PDF) file. This file is then sent to the customer via the Internet and is viewed on the computer screen of the customer. In the preferred embodiment, the preview is displayed as a PDF file. While other types of files might be used (GIF, etc.) PDF files are preferred because first, they are extremely high in resolution quality, and second, a PDF file  
20       provides a customer with a well-known format to process and view the preview layout.

      The customer then views the file and determines approval (or not) of the item. If the customer desires to change their individual data, the customer then views the order form again, changes their data, and the system generates a new preview file. If the item is approved, the customer clicks a button that tells the system to save the order. The order  
25       data for the customer (i.e. quantity, shipping address, etc.) is saved to a back-end database, and the Print Ready File is saved on a server. Once the order is saved, it may be tagged as a "pending" order or a "released" order. (Some customers wish for all of their orders to be stored in a holding queue so that an administrator may grant them final approval. These are considered "pending" orders. Once the administrator grants final approval, the "pending"  
30       order is marked as a "released" order.)

      Once an order has been released, it goes through the various stages of the production process (e.g. setup, composition, imposition, etc.) which are described in further detail below. Each stage of the process uses the Print Ready File that was generated when the



user created their preview. This file remains unaltered all the way through the printing process. Once the order is printed, it is shipped to the customer, and the order is complete.

Referring now to Figure 4, further system-level details of this overall process are shown. A block diagram 400 is shown of the Print Ready File system and the interaction of representative components. In general, this Figure describes an overview of an Internet-based ordering system (as stated above, other ordering modes might be used). The customer 402 is shown interacting with a customer computer 404. A website residing on the primary webserver 408 is contacted via the Internet link 406. An Image Logic Information Database (ILIAD) 410 is coupled to the server 408.

The general data composition of ILIAD 410 is further described in Figure 4a. The elements shown are meant to be illustrative, and are not meant to limit the data structure of ILIAD to such elements. Product and design information are shown generally as element 460, and is shown to further include asset information 462. Asset information is intended to include various customer logos, text, or fonts (i.e. "assets" of the customer) to be used on the printed products. Such information might be provided as data files, or via menu prompts and the like, from the customer. Specifications and costs 464 would include information pertaining to individualized costs for implementation of certain designs, and the like. Layout rules 466 would include the various rules to be used in arranging figures or text on the printed product, so that conflicts and inappropriate layout schemes do not occur. Customization rules and options 468 might provide for further custom design capabilities in arranging unique layouts.

ILIAD 410 is also shown to include manufacturing information 470. Such manufacturer information might include (but is not limited to) imposition rules 472, separation rules 474, vendor information 476, and trapping rules 478. These various rules are used in the production engine for arranging and preparing the images and/or elements in the Print Ready File (PRF). Order processing and work-in-progress (WIP) information 480 is also shown. Such information might include (but is not limited to) customer information 482, work orders 484, shipping information 486, and pricing information 488. An IOPC (ImageX Online Printing Center) database 490 is shown incorporated with ILIAD, with further details regarding its data contents described in Figure 4b. The IOPC database might also exist separately from the ILIAD, but is shown incorporated here as one embodiment.

Referring now to Figure 4b, the IOPC 490 is shown to include (but is not limited to) corporate procurement rules 491. Such rules might further include users/roles 492,



privileges 493, purchasing rules 494, and billing/shipping rules 495. Customer Products/Assets 496 are shown further comprising IOPC 490. This data grouping 496 might include design/brand information 461, asset information 463, catalogs-products-kits 465, and customization rules/forms 467. The IOPC 490 is shown to further include a  
5 variable information database 497. This data store contains information that regularly changes, such as locations, departments, titles, etc. 469. Employees 471 are also included in this data grouping 497.

Referring again to Figure 4, the PRF 412 is next sent to the Farm 414 (or "the Farm"). The Farm 414 is generally comprised of at least one, and usually several, high  
10 powered computers (e.g. a PC running Windows NT). The farm is designed to load balance file processing tasks by determining system impact of various jobs and distributing them accordingly. The Farm is also highly scalable, with control being routed through a single point of contact (i.e. a server, which might be referred to as a "Master Farmer"). Each different file processing module (or "Farm Plot") runs out of process from The Farm main  
15 process. Within the Farm, each Plot is controlled by a "Field" which is specific to the plot. The Field communicates with the Plot and handles all the specific interactions with the Plot. Jobs can be re-routed if failures occur within any particular Farm, Field, or Plot. Time estimates can also be provided regarding the processing of jobs. The Farm, in general, introduces little overhead in processing of tasks, and each different Farm service can be  
20 configured to run any of the file processing tasks. The Farm 414 provides a platform apart from the webserver 408 for running processing steps on the PRF. It should be noted that any such processing could also be done on the webserver 408, with load balancing of the job processing managed there.

The completed PRF 416 is thereafter passed onto the Asset Management File Server  
25 (AMFS) 418. The general data composition of the AMFS is further described in Figure 4c. The AMFS 418 is file server (or database or the like) used to store components relating to a client's product which should generally not change. In other words, these are the "Assets" of the client, such as company logos and the like. Such components are intended to include (for example) Encapsulated PostScript (EPS) files containing customer logos and graphics.  
30 Further included are diagrams, illustrations, static text and the like. Referring now to Figure 4c, the AMFS 418 is shown to contain representative data, including for example low resolution EPS files 419, high resolution EPS files 421, Preview PDF files 423, and PostScript Fonts 425.



Referring again to Figure 4, the user can also request a preview of the PRF 420. The Farm 414 reads back the preview PRF 422 from the AMFS 418 data store. The preview PRF 422 is then sent back to the web server 408 which applies software such as the Adobe Acrobat Distiller program. This (or similar) software reads the PRF and creates a PDF or similar file. The preview PRF file 422 is then sent to the user via the Internet and is viewed on the customer's computer screen.

If the preview PRF is accepted by the user, the finalized PRF 424 is thereafter retrieved from the AMFS 418 and sent for further processing operations. A batcher 426 and plater 428 are shown which are each typically comprised of a PC or the like. The batcher 426 receives the PRF 424 and performs logical imposition on the data. This would include server based software for automatic imposition. The plater 428 performs further steps including, for instance, imposition and color separation, and the formation of a high resolution print file. Both the batcher 426 and the plater 428 communicate via link 411 with the ILIAD 410 in order to read and use the rules stored therein in performing their designated tasks. The batcher 426 and plater 428 also communicate via link 427, which might include an TCP/IP link or the like.

A plate file 430 is thereafter stored in the AMFS 418. The plate file 430 is also sent to a vendor order system (VOS) 432. The VOS 432 is typically comprised of a PC or the like. The VOS 432 serves as a transactional machine, or a gate for all other vendors which might exist downstream. The VOS 432 might process tasks or information, including but not limited to, job instructions, purchase orders, invoices, payments, and shipping status of orders. The VOS 432 includes a link 434 to the ILIAD in order to retrieve various business information pertaining to particular customers. The VOS 432 receives a plate file 430 from the plater 428. In this example, the plate file 430 is yet another type of PostScript file (whereas other types might be used).

Thereafter, the elements and steps which are shown are illustrative of what might occur after the VOS 432. The VOS 432 might be used to send the consistent plate file to any other system or request source via any reasonable medium. Such information could be (for example) traded, auctioned off, or distributed across many different markets, in many different ways, and across many different mediums. It could be supplied by various customers and aggregated for processing by VOS and ILIAD. In this example, an Internet connection 436 is shown wherein a vendor computer 438 interacts with the VOS 432. The vendor computer 438 negotiates an order with the VOS 432 and receives the plate file 430.



Many other such vendor computers might exist and contact the VOS 432. Vendor computer 438 thereafter sends the plate file 430 to a Raster Image Processor (RIP) 442. Note that the plate file might alternatively be sent directly to the RIP via link 440 if the VOS 432 is not a desired element in the process. The RIP 442 is typically a PC or the like running RIP software. The RIP produces a bitmap file 443 which is sent to a Recorder 444. The recorder 444 is an image setting device which takes the raw bits from the RIP and translates them into a press input medium 446. Such media 446 might include film, RC paper, or whatever input source the press 448 is looking for. The press 448 takes the input medium source and produces the end result, in this case a business card 450. The business card 450 is shipped or routed 452 back to the customer 402 to complete the overall process.

The overall process 400 described in Figure 4 makes use of an inventive Print Ready File that provides many advantages, including but not limited to the following: (1) The file preferably maintains its state, even if the elements that comprise the file are altered in the creating application. For example, high resolution files can be logically embedded for access and processing at an appropriate point in the Pre-press process described herein. (2) The file contains both elements that the customer wants to see in a preview, and elements that a print vendor needs to see for a printing, each one being potentially mutually exclusive. (3) The file can be fed, or converted and fed, into imposition software, and color separation software. (4) The file can be fed, or converted and fed, into software to generate a PDF file or bitmapped image for a preview. (5) The file can create basic elements (e.g. ten, and small in file size) in a relatively small amount of time (e.g. under a second) on a desktop hardware platform. (6) The file supports three representative element types: text, line, and embedded graphic (raster or vector). (7) The file preserves the integrity of embedded graphics which originate from customers using standard graphics programs such as FreeHand, Quark, Illustrator, PhotoShop and PageMaker. (8) The file maintains the integrity of text, line, and graphic elements in both print and preview. This integrity can be demonstrated in the way of fonts, kerning, leading, line widths, and graphical reproduction. One possible exception is that raster images may be downsampled when converting to a preview file type.

The output to the print vendor should preferably be in Level 1 PostScript, to support all possible RIPs. To accommodate these features, the preferred embodiment implements the Print Ready File in the Adobe PostScript language. It should be noted that other languages aside from PostScript can also be used that support the above conditions. For



example. other page composition languages/formats can be used. Also. other RIPs or specialized equipment can be supported for custom print orders, and the like.

Resulting features of the present invention include, but are not limited to, the following: (1) The system or site contains all of the data the customer needs in order to print the customer's materials. (2) Data are completely secure and is the property of the customer. (3) The system or site incorporates rules to ensure that no matter what party might happen to input data, the resulting printed materials are printed in a manner consistent with the corporate image and design policies that have been approved. (4) The system or site incorporates a variety of business logic, including procurement, authorization security and billing rules defined by the Company. These rules often set up an authorization process whereby an employee puts in an order and it is routed to the authorized party. The purchasing administrator might then release the particular order to be sent to the printer.

#### PRINT READY FILE

In order to accommodate the "state" of the file (e.g. Print or Preview), a global numeric variable, or "flag", is used to indicate which elements will image when this file is processed. In this example, this flag is a PostScript integer and will be used for bitmasking each of the state flags of the individual elements. Each element has four possible states: Print, Preview, Both, and none. This global flag is written into the PostScript stream such that an application can find the position of the flag within the file, and alter the value as needed. The default or original value of the flag will be set to include elements that are in a Preview or Both state. It is a unique and efficient aspect of this invention that a single flag may be used. However, alternative embodiments might use multiple flags, comparative elements, or runtime logic to evaluate the appropriate state and direct processing, all within the spirit of the invention.

The Line and Text elements will each check their respective state flag against the global flag to see if their bit values are contained within the global flag, using a standard bitmasking technique of a logical AND operator. If the elemental state flag bits are not within the global flag (a zero result from the logical AND), a function is utilized to move the PostScript interpreter's file pointer past the end of that particular element. The element is thereby skipped, and the element does not image (see, e.g., Figure 5b below).



Embedded graphics will use a different method than Line and Text elements for selecting Print, Preview, and Both states. When writing a graphic with a state of Preview, the graphic is embedded in the file, but OPI comments are used to link to a blank PostScript file. When using this file with a global flag set in a Print state, the blank PostScript file is used instead of the Preview one that is embedded within the file. When writing a Print graphic, a blank PostScript file is embedded, and OPI comments are used to link to the graphic. When writing a graphic that is in the Both state, the graphic is embedded with no OPI. There is a caveat to graphics in a Both state, and that is when the image is high resolution. High resolution raster graphics are usually very large. One purpose of the present system is to create a file that is relatively small, thereby enhancing speed in working with the file. Here, the OPI comments are used to facilitate a solution. The low-resolution version of the graphic is embedded in the file, and the OPI comments are used to link to the high-resolution version. In this state, when using the file for Preview, the low-resolution graphic is seen. When using the file for Print, the high-resolution file is used. Because of the OPI implementation, the programs used for creating Previews of the PostScript file are preferably configured to remove the OPI comments. The programs that utilize the PostScript file in a Print state, should preferably be configured to process the OPI comments.

An important feature of the present invention are the OPI links to external documents. Along with the Print Ready File, each of the externally referenced files need not change. This is implemented, in part, by securing the directory where the graphics reside, and using operating system security. Only applications controlled by the present system might be used to add files to this directory. These applications might not allow the modification or deletion of any of these files, but only the adding of new files. In this manner the externally referenced files are secured such that they cannot be altered by accident, or on purpose. They can also be secured by access codes or authorization, as between print and preview modes.

Referring now to Figure 5a, a representative construct of the Print Ready File 500 is shown. A global flag 502 is used to indicate which elements can print. This flag is numeric and is used to apply a bitmask to the elements. For example the value "0 1" indicates that the elements only in the "Preview" state will not print, while those in "Print" state should be printed. In this example, it is shown as a 32-bit PostScript integer. Additionally, shown is a text element 504, a line element 506, and a graphical element 508. Each text and line



element has associated with it four "states": Print, Preview, Both, and none. However, the graphical element does not use the "none" state. Preferably, these states of an element are represented in a 32-bit integer, similar to the global flag, termed a "state" flag.

5 The text and line elements 504 and 506 will each check their respective state flags against the global flag to see if they should be imaged. If the text or line element state flag does not match with the global flag, then the present invention will apply a routine of PostScript operators to move the interpreter's file pointer past the end of the element in question, thereby skipping that element such that it does not image. For example, if text  
10 element has its "Preview" bit set, it would still not be imaged during a preview unless the "Preview" bit of the global flag was also set. This routine, hereafter referred to as "Global Flag Comparison Logic" is shown encompassing the text element 504 with a start function 510 and an end function 512. The same logic is also shown encompassing the line element 506 with a start function 514 and an end function 516. Each element in the file preferably has this logic wrapped around it.

15 Figure 5b shows a flowchart 520 of the Global Flag Comparison Logic. Process 522 shows that for each text and line element, the state flags of the element are compared to the global flag in 524. If the global flag matches the state flags, then the element is processed in 526. If the global flag does not match the state flags, then the element is not processed. The preferred embodiment skips the element by moving the pointer past the element via a  
20 PostScript routine. The Global Flag Comparison Logic then loops back via 528 until each element is completed.

Embedded graphic elements will use a different method depending upon the setting of the Print, Preview, and the Both state flags. The Print Ready File is being passed from point to point. In general, it is desired to keep the size of the Print Ready File down to a  
25 minimum for certain operations, thereby increasing the efficiency of the overall system. This is done by directly embedding a low resolution graphical object into the file for preview operations. For print operations, the preview graphic is removed and a link to a high resolution graphic is supplied instead. When the flag is set for "both," then a low resolution graphic is embedded in the file, and a link is provided to a high resolution  
30 graphic.

Figure 5c shows the elements of a representative Print Ready File structure 530 (with certain similar elements numbered as per Figure 5b). In this example, the Preview flag is set for the graphic element 508. When writing a graphic for preview only, the



graphic element will actually be embedded as, for instance, an EPS graphic element. The file 530 uses Open Prepress Interface (OPI) comments 532 (shown as OPI start) and 534 (shown as OPI end) which are placed around the embedded graphic element 533, and an OPI link 531 to a blank extended PostScript file 536. The process of OPI takes an  
5 embedded EPS file and replaces it with an external EPS file. This replacement is done by software which processes the OPI comments. If a PostScript processing software (such as the one used for previewing a file) does not process OPI comments, then the embedded EPS files are viewed. If such comments are processed, the present system utilizes a property of the OPI code -- which generally needs to find a link to a high resolution graphic (blank or  
10 otherwise). The OPI comments are thereby generally used for designating preview and print EPS files. It should be noted that other "comment code" or the like, could also be used to direct the workflow.

Figure 5c again shows a preview mode example. If an EPS file is set for preview, then it is embedded directly in the PostScript file. A link is still provided for the OPI code -  
15 - however, this link is to a blank file. Hence, a preview operation (when "preview" of the global flag is set) will show the embedded graphic element. However, when using this file during a print operation (where the OPI is processed), the blank PostScript file is used instead of the preview graphic element 533 that is embedded within the file. In this fashion, the preview EPS image is removed from the final printed image.

20 For a print operation, an opposite tactic is performed. A blank EPS file is embedded in the Print Ready File, with an OPI link to the actual print EPS file. When previewing, no print image is shown. The embedded blank image is shown, which effectively shows nothing. When printing (where the OPI is processed), the blank EPS is replaced with its linked print EPS file. In this manner, the print EPS file is added to the final image.

25 For example, when writing a Print graphic only, Figure 5d shows a representative Print Ready File structure 550 (with certain similar elements numbered as per Figure 5b). In this example, the print flag is set for the graphic element. A blank PostScript graphic element 556 is embedded in the file. The OPI comments 552 (start) and 554 (end) are used via the OPI link 558 to link to the graphic 560. In this instance, the printed graphic is  
30 shown as the print EPS file 560.

When writing a graphic that is in the Both state, Figure 5e shows a representative Print Ready File structure 570 (with certain similar elements numbered as per Figure 5b). In this example, the Both flag is set on the graphic element 576. In this situation, two files



are used, a low resolution file and a high resolution file. The low resolution graphic element 576 is embedded directly in the file structure 570. An OPI link 578 is used to link to an external high resolution EPS file 580. As discussed above, the OPI start 572, OPI end 574, and linking comments there between, facilitate this linking operation. Hence, when  
5 using the file for Preview, the low-resolution graphic is seen (as the OPI comments are not processed). When using the file for Print, the high-resolution file is used, as the high resolution EPS file is linked in per the processed OPI comments. Note that if the high resolution file is small enough, then it too might be embedded within the PRF and no swapping operation would need to be performed. In this case, the bitmask will be used.  
10 Otherwise, the described use of OPI links provides a smaller and more usable Print Ready File.

As a result of this implementation, the preferred embodiment would use programs for creating Previews of the PostScript file that are configured to remove the OPI comments. The programs used for processing the PostScript file in a Print state should  
15 preferably be configured to process the OPI comments. The global flag will be written into the PostScript stream such that an application can find the position of the flag within the file, and alter the value for its present needs. The default or original value of the flag will be set to include elements in that are in a Preview or Both state. These operations are described in further detail below in association with the flowcharts for certain applications.

20 Prior to creating the output file, the system might provide the user, through various common means, a way to mark the elements as Print, Preview, Both or none. This state can be added to a current system that is used to create printing products, or integrated with, or translated from, systems in which the state is static, or in which it is set by program logic. The user will be able to select the state for each element within the product. The application  
25 creating the PostScript file will then need to use the provided state for comparison against the global flag.

### BUSINESS CARD EXAMPLE

Figures 6-8 illustrate an example of a business card at various stages of the process.  
30 Referring now to Figure 6, an example business card 600 is shown as a result of a preview operation by a user. The Figure is in PDF format, and might be viewed with any PDF viewer, for instance Adobe's Acrobat Exchange v3.1. In this example, the logo 602 (which



might include colors) and the "COMPANY Worldwide" text 604 below is an embedded graphic with its state flag set to Preview. The text blocks 606 and 608 have state flags set to both. Any other elements which are not shown are graphic elements with the state flags set to Print. These Print elements will be visible at the Imposition stage.

5           Figure 7 shows the example Preview file 700, when imposed (see description below). Figures 8a, 8b, and 8c show the respective color separations 810, 820, and 830 for the preview file (also further described below). Embedded graphics that were in a Print state are now visible. For instance, two embedded graphics might show up on separations called emboss (830) and deboss (820). These two separations are used for creating special  
10       dies that make impressions in the final printed product, such that it either "sticks out" (emboss) or "depresses in" (deboss). At this stage, a file containing multiple Print Ready Files is ready to be sent to a RIP and image setting device.

#### PRINT PRODUCTION PROCESS FLOW

15           In order to take advantage of the unique data in the Print Ready File format, certain applications are needed. The applications have knowledge of the format, and are capable of utilizing the data contained therein. Different PRF applications are used at different stages of the print production process. Below, an overall flow of the process steps are first described. Thereafter, certain stages of the print production process are described in further  
20       detail. While described as flowchart steps, it is generally recognized that persons of ordinary skill in the art will recognize how to implement such applications from the flowchart descriptions.

          Figure 9 shows an overall flowchart 900 of the print production process. In step 902, the user first provides business information to a person responsible for setting up the  
25       user account. This first step involves considerable human interaction, but the step generally needs to be done only once in order to properly set up the account. Such information might relate to: purchase orders, authorizations, employee information, employee lists, product styles, style guides, samples, graphical information and fonts. Products would include items such as business cards, envelopes, stationery and the like. Step 904 involves a customer  
30       setup application, wherein the elements within a business card or product are defined and stored. Customer setup 904 is described in further detail in Figure 10. Step 906 involves the customer providing information regarding the product by using the customer website



referred to in Figure 10. Once such information has been entered, the system can perform the composition step in 908. Composition creates the Print Ready File according to steps further detailed in Figure 11. Generally, the user will request a preview file in step 910 in order to view the results on-screen before printing. The steps involved in the generation of the preview file are further described in Figure 12. The decision block 911 sends the user back to step 906 if further changes are desired. If the preview file is acceptable, then the order is placed in step 912. Thereafter, the process of imposition (or batching and plating) of the PRF is performed in step 914. The imposition steps are further described in Figure 13. Color separation 916 is next performed, with steps described in further detail in Figure 14. Color separation produces color separation plate files which are transported to the print vendor in step 918. Step 920 shows the processing of the color separation plate file by submitting the file to a Raster Image Processor (RIP). The RIP generally produces a bitmap file which is converted into the printed product 922. The product is thereafter shipped to the customer in step 924.

Figure 10 shows certain more detailed steps associated with the customer setup application 904 from Figure 9. In the setup process, product setup software is used to create each of the basic elements, and associate a state flag with each one. This software therefore identifies the position, size, contents, etc. of each element type. Step 1002 is the process of determining the printing requirements of a product. This is generally done via a human specialist interacting with the customer to gather and generate graphic and textual materials pertaining to that customer. In other systems, tabular layout or other document definitions are used to gather and create the derived graphic and textual material (as in XML-based processing of data and Document Type Definitions). Other steps might include die creation, and other related procedures. Step 1004 next is the performance of the Prepress process. This typically consists of generating and verifying the customer assets (e.g. EPS files and fonts). These assets are then added to the Asset Management File Server (AMFS 418 as per Figure 4), or other such database.

An EPS is a file format used in Prepress operations, and generally contains information required to create a printed document containing graphics images. Along with the imaging bits, EPS files contain other data respecting reproduction of the image for digital display, or for print, such as color selections, color settings, scaling of graphics, embedded fonts and so forth. Such files often need to be "Washed" or converted into a consistent format for automated processing. Washing is one of several Prepress operations



that can be automated by hosting the application on a server, or other networked computer, and maintaining control of certain operations as part of a distributed Prepress software operation.

5 Other Prepress operations that can be automated (in a similar fashion) include, but are not intended to be limited to, creation of Print Ready Files, trapping, color separation, and imposition of Print Ready Files.

In step 1006, the user is further prompted for information regarding the product, as might be needed for a particular print job. Step 1008 shows the process of defining the composition rules for each of the particular elements. Step 1010 further shows that for each  
10 element, the x, y, and z position of the element in the product is defined. In step 1012, a type and state is assigned to each element. The "type" includes line, text, and graphical, whereas the "state" includes Preview, Print, Both, or none. Step 1014 next shows the assignment of various other properties (as needed) to each of these elements. Once finalized, this information is stored via step 1016 in the rules database (or ILIAD 410 as per  
15 Figure 4).

A customer website is created in step 1018. The customer accesses the website to provide various customer information. The user is prompted for information in step 906. Text elements might require a prompt, in that the user is providing textual information in response to a question. Line and graphical elements, however, might be retrieved directly  
20 from the appropriate database via a locator, index, or the like. Once the user enters the requirement information, it is stored in the ILIAD database as per step 1020.

The next stage of the process involves composition of the Print Ready File. When created, the Print Ready File in the preferred embodiment follows PostScript conventions including, for instance, DCS standards, platform independent operators, and color  
25 separation functions. The file might also include a bounding box, which is not required for a multi-page PostScript file, but might be used by other applications in the process to identify the size of the image to be rendered.

Referring now to Figure 11, a more detailed flowchart is shown of the composition process 908 from Figure 9 (with additional references to elements from Figure 4 in  
30 parenthesis). In step 1102, the web server (408) requests the PRF from the Farm (414), along with related user information. In step 1104, the rules regarding the product setup are read from the ILIAD (410) database. The global flag is next written into the PRF with a



default setting of "Preview" as shown in step 1106. A loop is then performed for each element of the product 1108. The element information is retrieved, e.g. data source, component data, and state. Based upon this information, and the logic described above, the element is written into the PRF in step 1112. The loop continues via link 1114 until each  
5 element of the product is completed and written into the PRF. When finished, the PRF is stored in the Asset Management File Server (418). An alternative embodiment could substitute receipt of one or more data streams in response to the web server request in step 102, as with XML output from one or multiple machines performing the required pre-press operations. The rest of the operations described proceed as depicted. Once the PRF is  
10 created, a preview file is generated as will now be explained, and then summarized in Figure 12. An example preview image of a business card is shown in Figure 6. Since PostScript is a page description language, a PostScript interpreter is used to process the Print Ready File which is generated in PostScript. In order to preview such, the Print Ready File is preferably first interpreted and then rasterized. Rasterizing is the process of  
15 interpreting the PostScript file and converting it into raster image data, or pixels for a bitmapped image. As the target audience is a human previewer, it is desired to convert the Print Ready File to a format that will allow the user to view the data. The process of rasterizing PostScript fixes the image into a static bitmapped image. One drawback of this format, however, is that the static image does not allow for "zooming" operations. In other  
20 words, the static image also does not allow for greater detail at higher zoom factors, since the image is already a fixed bitmapped resolution. One preferred embodiment therefore uses a format that supports vector images and can be selectively scaled and rasterized on the fly when presenting the data to a user. PDF is an example of a format that allows for rasterization on the fly by using a vector source file (with the option of embedded pre-  
25 rasterized images).

A PDF file can be created directly, or by using tools which convert PostScript to PDF. Adobe's Acrobat Distiller is an example of one such tool. The settings on these tools are an important consideration. With the present implementation of OPI, the OPI comments  
30 are embedded in the file, and should pass through without OPI replacement, or external links. Because PDF is a page description language that is devoid of logic, all of the related state flags are processed in the creation of the PDF file. Distiller has a PostScript interpreter built in, and instead of rasterizing the PostScript, it generates PDF operators. Since it is a



PostScript interpreter, it can process each element using the aforementioned Global Flag Comparison Logic (see Figure 5b)

Other settings are also important for the creation of a preview PDF file. These settings include, but are not limited to: raster image compression, raster image  
5 downsampling, and font subsetting.

Regarding raster image compression, many embedded graphics contain both vector and raster data, and since the goal is to have a very small preview file, it is desired to compress the raster image data, but not to the point of distortion. PDF allows for images to use lossy or non-lossy compression schemes. The present system chooses a non-lossy or  
10 lossless compression scheme. Lossless compression involves no image information being removed from the file for the sake of compression. Lossy compression removes image information to achieve higher compression rates. Note, however, that lossy compression is an alternative embodiment appropriate if high assurance of image quality is not an issue for the customer or nature of the printed document, and in the case of electronic display of  
15 images, is of little consequence by comparison to limits on display resolution.

Regarding raster image downsampling, each embedded graphic element might be created via some form of setup software that allows the user to downsample the image. Downsampling takes the file and converts it to a lower resolution. This process can distort images, by reducing the number of pixels in the graphic. Since the present system allows  
20 the user of the Product Setup Software to determine the setting, this allows for a maximum of image quality to be retained. Computer display monitors have a significantly lower resolution than imagesetter devices; A standard resolution for monitors is 72dpi, whereas a standard resolution for imagesetter devices is 2400dpi. As a result, the image is preferably downsampled on the fly by the PDF reader for display to the user on a monitor. An  
25 optimum downsampling will have little affect on the visible quality of the image when viewed on screen.

Regarding font subsetting. Since they contain vector data, PDF files do not rasterize the text when created. Text is rasterized by PDF reader software when displaying the PDF file to a user. Because the rasterization happens when viewing the file, the fonts used by the  
30 text need to be included in the file. Fonts are not small, and one aspect of our preview file is its relatively small file size. As a result, Distiller contains options to subset fonts. The basic process of subsetting a font is to remove all characters from the font that are not used



in the document. This greatly reduces the size of the font in the file. The present system teaches the use this option for preview files.

There are many other settings for PDF files that can optionally be used. These are not a requirement of the process, but can affect file size and display performance. These options include conversion of CMYK colors to RGB colors, converting Device Independent color spaces to Device Dependent color spaces, and applying transfer functions. Other forms of image management and substitution, such as Kodak Flashpix data structure, could be adopted without changing the basic convention. None of these options, however, detract from the desired features of the Print Ready File.

The present system might also use a bitmapped preview implementation. There are many products on the market for creating bitmaps from PostScript files, and most fall into the category of a Software RIP. One such product is Image Alchemy by Handmade Software, Inc. Given that the Print Ready File typically contains all needed fonts, it is possible to avoid font substitution. Instead, the present system simply provides the output dots-per-inch (dpi) and the graphic file format. Alchemy does not support OPI, so the comments are ignored, and the embedded preview graphics are used. Bitmaps are simply collections of bits, and hence have no logic built in. As a result, all of the state flags should be processed in the creation of the bitmap file. Given that a software solution like Alchemy is a PostScript interpreter, it follows the same rules as used for creating a PDF, following the Global Flag Comparison Logic for excluding Print elements.

Referring now to Figure 12, a flowchart is shown of representative steps associated with the element "generate preview file" 910 from Figure 9. Figure 4 elements, when referenced, are shown in parenthesis. If the user desires to preview the file, the web server (408) requests a preview file from the Farm (414) as shown in step 1202. The Farm then retrieves the PRF from the Asset Management File Server (418) in step 1204. The Farm sets the global flag in the PRF to preview mode in step 1206. In step 1208, the Farm converts the PRF to a preview file. This is done via a PostScript interpreter which results in common image formats using the Global Flag Comparison Logic (and the OPI comments). Common image formats include, for instance, bitmap and PDF. In step 1210, the preview file is thereafter sent to the web server (408). In step 1212, the user then accesses and views the preview file via a web browser or the like.

Just prior to the imposition process, the global flag within the Print Ready File is changed to one representing Print only. Once the flag is flipped, the file is ready to be



imposed into a file with many other such Print Ready Files. Imposition software takes several PostScript files and creates a single file with all "imposed" files embedded, adding crop marks, registration marks, and the like. Figure 7 shows a representative example. Regarding this image, a few things should be noted: the text is still there; there are crop marks on the top left, top right, bottom left, and bottom right; the word Composite is describing the type of image as opposed to a color separated view; the logo has changed. What is shown is comprised of two embedded graphics on top of each other, each one for a different purpose, as shown during Color Separation.

During imposition, it is important that the OPI comments that were generated are so processed, and the embedded "preview" graphics are replaced. The options for the software used in Imposition should be set to force the processing of OPI. The preferred embodiment implements this step using software such as Preps by ScenicSoft which has an option called "OPI Force Merge." This option is selected and the image is generated.

Referring now to Figure 13, a representative series of steps describing the imposition process 914 from Figure 9 is shown. Figure 4 elements, when referenced, are shown in parenthesis. The batcher (426) performs logical imposition according to standard techniques. In step 1302, the plater (428) reads the imposition requirements from the ILIAD database (410) for the product, and for the vendor, as needed. Step 1304 shows that for each PRF in the plate a loop is performed. In step 1305, the PRF is retrieved from the Asset Management File Server (418). In step 1306, each global flag is set to "Print" mode. The PRF is then written to a composite plate file in step 1308. The composite plate file is thereafter stored in the Asset Management File Server (418) in step 1310.

The color separation phase takes the imposed, OPI processed file, and generates "separations". Separations are either represented in a single file, or multiple files. One software implementation which might be used includes Preps by ScenicSoft which generates a single file with all "separations" therein. This is the final stage of the PostScript file prior to being processed on a RIP. Accordingly, all necessary settings for the destination RIP are specified in this stage. The destination device is provided to Preps, which uses an Adobe PPD file to generate commands so that the destination device can understand the settings for page sizes, emulsion, polarity, and other such options. The PPD file is a file provided by the Imagesetter vendor that provides all the details necessary to generate a device-specific file. As further described above, Figures 8a-8c show the color separation stages of an example file.



Of further note, the aforementioned color separation software (Preps) has an option to deal with missing fonts. Since the Print Ready Files according to the present invention should contain all the fonts necessary to produce the file, this option to supply fonts is turned off.

5 Referring now to Figure 14, a representative series of process steps is shown for the color separation element 916 from Figure 9. . Figure 4 elements, when referenced, are shown in parenthesis. In step 1402, the composition plate file is retrieved from the Asset Management File Server (418). The plater server (428) then converts the composition plate file into a color separated plate file in step 1404.

10

### ALTERNATIVE EMBODIMENTS

In the primary implementation, a PostScript file format is altered and used to store additional information about a product which allowed the system to use that file in all stages of the production process. Alternate implementations could use a different file format to  
15 achieve this goal. For example, the system might use the Portable Document Format (PDF) to store this information. PDF is similar to PostScript in many respects, and could easily be modified to fulfill the objectives of the present invention.

The primary implementation also uses a "state flag" or "tagging" scheme to identify certain images as preview or print images. Any page description language which includes  
20 tagged elements would be easily converted and used with the present system. It would also be possible to modify other file formats that do not use tagging, such as TIFF. While these formats do not inherently provide the facilities needed to provide the described Print Ready File output, certain modifications could be made to provide such features.

It is also possible to create a unique file format that satisfies the functionality of the  
25 present invention. PostScript was chosen due to its widespread support in the printing industry. While use of Level 1 PostScript has been described above, Level 2 and/or Level 3 (and/or subsequent versions) of PostScript might similarly be used. Moreover, there should not be any technical problems or limitations associated with creating a new and unique format (i.e. other than PostScript) to implement the Print Ready File output system  
30 described herein, or in translating the image files between formats.

The original PostScript solution could also be programmed in other ways to provide slightly different solutions. For example, the primary implementation describes the addition



of order data to the Print Ready File in text format. It is possible to add this data to the file in a format such as XML (extended markup language). This is a format that allows a document to contain both data and a description of that data. XML would allow the data to be read and processed by any number of external databases, viewers, web browsers, and other such applications. A file with this information could be seamlessly integrated into the system to provide printing, billing, and shipping services automatically, based solely on the content of the consistent PostScript file, as referred to as the Print Ready File above. XML is a derivative of SGML (Standard Generalized Markup Language). There are many other SGML derivatives that might also serve this purpose.

Of further note, the tagging scheme used by Print Ready File allows for four different states for an image (e.g., preview only, print only, both, and none). The Print Ready File system actually has the ability to handle any number of states. Because this is a software system, it is possible to alter the system at any time to add new states, or to dynamically vary the state record, type, or the number of state descriptors. (See below for an example of an alternate workflow that could use this feature.)

Another technique might be used in the subsystem that actually reads the tagged data and images it. In the primary implementation, the entire Print Ready File is transmitted throughout the system at all times. However, once the file has been sent to the vendor, the entire file may no longer be needed. For example, a vendor might use a RIP to process the PostScript files and then send them to an imagesetter, and then to a press. In this case, the vendor's system only uses the print layout and has no need for the preview layout. While it is still possible to send the preview layout as part of the file, it might not be as efficient, especially if the system is low on processing power or network bandwidth.

In such a case, the system could be programmed to detect if and when certain portions of the file were no longer needed. If such a situation were detected, the system could send the Print Ready File through a preprocessor, which would cut out the unneeded portions of the file and send the remainder to its destination. Note that the original Print Ready File would still be maintained on a central server. The server would be responsible for figuring out when and where it would be safe to send smaller files. This would still maintain the consistency provided by the original Print Ready File; the system is merely reducing the amount of data it transmits for optimal efficiency.

The above embodiments have described a workflow in which the preview layout of the Print Ready File is viewed by a client through a web browser and the print layout is used



by the vendor to produce the item. There are many other workflows in which Print Ready File could be used. A list of several alternate workflows includes, but is not limited to, the following:

5       The customer does not have to be the only one to use the preview layout. In a printer specific workflow, the Print Ready File can be sent to a digital proofing device and/or a computer screen that has been optimized for color correctness. The same file can then be sent to a high-resolution imaging device for creation of the plates or directly to a digital press. The vendor system might be configured to automate this process based on the content of the Print Ready File.

10       Previously, the possibility was raised of a format whose tagging scheme allowed many different states. As an example of how this might prove to be useful, an advanced online ordering system is considered. This system provides a user with the ability to select one imprint (e.g., a company logo or advertisement) and then choose to have that imprint imaged on a number of different products at once (e.g., posters, coffee cups, T-shirts,  
15       embroidered hats, etc.). It is possible that different products might require slightly different versions of the imprint image. For example, an embroidery machine might not be able to display complicated gradients, so it might require a version of the image in which gradients have been converted to blocks of solid color.

20       In such a system, the Print Ready File could contain all of the image versions required for all of the products. The order might contain a list of all of the products that are to be produced, and as the printing system imaged the order for each product it might get the appropriate image data from a single Print Ready File.

25       A similar workflow might be used in a publishing scenario. A customer might want to produce copies of a document both on paper and on a CD-ROM. In this situation, the product would use different images and different production information, but might also use a completely different production method or vendor. The system might read the tagged information in the file and send one job to the press to be printed and another job to a CD production house to be pressed onto a CD.



The number of products that could be manufactured using this system are numerous. For example, virtually any printed item could be created, including but not limited to: coffee mugs, T-shirts, hats, embroidered clothing, magnets, mirrors, toys, and any form of digital output display, and so forth.

5

## COMPUTER SYSTEM EMBODIMENT

Figures 15A and 15B illustrate a computer system 1500 suitable for implementing embodiments of the present invention. FIG. 15A shows one possible physical form of the computer system. Of course, the computer system may have many physical forms ranging from an integrated circuit, a printed circuit board and a small handheld device up to a huge super computer. Computer system 1500 includes a monitor 1502, a display 1504, a housing 1506, a disk drive 1508, a keyboard 1510 and a mouse 1512. Disk 1514 is a computer-readable medium used to transfer data to and from computer system 1500.

FIG. 15B is an example of a block diagram for computer system 1500. Attached to system bus 1520 are a wide variety of subsystems. Processor(s) 1522 (also referred to as central processing units, or CPUs) are coupled to storage devices including memory 1524. Memory 1524 includes random access memory (RAM) and read-only memory (ROM). As is well known in the art, ROM acts to transfer data and instructions uni-directionally to the CPU and RAM is used typically to transfer data and instructions in a bi-directional manner. Both of these types of memories may include any suitable of the computer-readable media described below. A fixed disk 1526 is also coupled bi-directionally to CPU 1522; it provides additional data storage capacity and may also include any of the computer-readable media described below. Fixed disk 1526 may be used to store programs, data and the like and is typically a secondary storage medium (such as a hard disk) that is slower than primary storage. It will be appreciated that the information retained within fixed disk 1526, may, in appropriate cases, be incorporated in standard fashion as virtual memory in memory 1524. Removable disk 1514 may take the form of any of the computer-readable media described below.

CPU 1522 is also coupled to a variety of input/output devices such as display 1504, keyboard 1510, mouse 1512 and speakers 1530. In general, an input/output device may be any of: video displays, track balls, mice, keyboards, microphones, touch-sensitive displays, transducer card readers, magnetic or paper tape readers, tablets, styluses, voice or



handwriting recognizers, biometrics readers, or other computers. CPU 1522 optionally may be coupled to another computer or telecommunications network using network interface 1540. With such a network interface, it is contemplated that the CPU might receive information from the network, or might output information to the network in the course of performing the above-described method steps. Furthermore, method embodiments of the present invention may execute solely upon CPU 1522 or may execute over a network such as the Internet in conjunction with a remote CPU that shares a portion of the processing.

In addition, embodiments of the present invention further relate to computer storage products with a computer-readable medium that have computer code thereon for performing various computer-implemented operations. The media and computer code may be those specially designed and constructed for the purposes of the present invention, or they may be of the kind well known and available to those having skill in the computer software arts. Examples of computer-readable media include, but are not limited to: magnetic media such as hard disks, floppy disks, and magnetic tape; optical media such as CD-ROMs and holographic devices; magneto-optical media such as floptical disks; and hardware devices that are specially configured to store and execute program code, such as application-specific integrated circuits (ASICs), programmable logic devices (PLDs) and ROM and RAM devices. Examples of computer code include machine code, such as produced by a compiler, and files containing higher level code that are executed by a computer using an interpreter.

As an invention implemented in software, there is inherent flexibility in creating the logic, system flow, and data structures necessary to program the invention. Data structures and values upon which calculations are performed may be explicit, derived from other data, imported from other sources, or result from program calculations or logical operations, all without departing from the spirit or limiting the scope of the invention. The algorithms in this patent, including the workflow integration, may also be expressed explicitly and/or derived without departing from the spirit of or limiting the scope of the invention.

Although the foregoing invention has been described in some detail for purposes of clarity of understanding, it will be apparent that certain changes and modifications may be practiced within the scope of the appended claims. For instance, the computer may function as a server or the like. Therefore, the described embodiments should be taken as illustrative and not restrictive, and the invention should not be limited to the details given herein but should be defined by the following claims and their full scope of equivalents.



What is claimed is:

1. A system for producing consistent visual medium materials comprising:  
a file including text, line and graphical elements representing the materials;  
a server device for interfacing with a customer computer;  
5 a first storage device which interacts with the server device and stores and retrieves rules regarding placement of the elements on a visual medium;  
a flag included in the file having certain states associated with each element, the flag and selected state being used to direct composition processing associated with each element;  
and  
10 a second storage device used for storing and retrieving static element information, wherein a Print Ready File results from the composition processing of the elements.
2. The system of claim 1, wherein the Print Ready File is a single file representing the composed elements.  
15
3. The system of claim 1, wherein the Print Ready File is used by a batcher device and a plater device which provides a plate file for use by a raster image processor.
4. The system of claim 3, wherein the batcher and plater device utilize the rules  
20 stored in the first storage device.
5. The system of claim 3, wherein the Print Ready File is stored on the second storage device.
- 25 6. The system of claim 2, wherein the Print Ready File is a PostScript file.
7. The system of claim 3, wherein the Print Ready File includes a global flag having at least a print and a preview setting.



8. The system of claim 7, wherein the state flag associated with each of the text and line elements includes at least a print, preview, both, and none state option.

5 9. The system of claim 7, wherein for each text and line element, the state flag options are compared to the global flag in order to decide whether or not to process each element.

10 10. The system of claim 9, wherein the element is not processed by moving the associated file pointer past the element.

11. The system of claim 8, wherein the state flag associated with each of the graphical elements includes at least a print, preview, and both state option.

15 12. The system of claim 11, wherein the Print Ready File includes executable code around the graphical element and associated links to facilitate processing according to the flag settings.

20 13. The system of Claim 12, wherein the executable code includes OPI comments.

25 14. The system of claim 12, wherein a setting of the preview state option causes a graphical element to be embedded in the Print Ready File, with a subsequent link to a blank graphical file.

15. The system of claim 12, wherein a setting of the print state option causes a blank graphical element to be embedded in the Print Ready File, with a subsequent link to a graphical file.



16. The system of claim 12, wherein a setting of the both state option causes a low resolution graphical element to be embedded in the Print Ready File, with a subsequent link to a high resolution graphical file.

17. A Print Ready File structure for producing consistent visual medium  
5 materials, the Print Ready File structure comprising:

a global flag having least a print and preview setting;

elements including at least text elements and line elements;

a state flag having at least print, preview, and both options associated with each  
element;

10 comparison logic code to compare the state flag of each element with the global flag,  
the element not being processed if the flags do not match; and

executable code to process the graphical elements according to their state flag  
settings, whereby the Print Ready File may be used to preview and to print the visual  
medium materials.

15

18. The file structure of claim 17, wherein a setting of the preview state flag  
option causes a graphical element to be embedded in the Print Ready File, with a  
subsequent link to a blank graphical file, whereby the graphical element is imaged during a  
preview of the visual medium materials.

20

19. The file structure of claim 17, wherein a setting of the print state flag option  
causes a blank graphical element to be embedded in the Print Ready File, with a subsequent  
link to a graphical file, whereby the graphical file is printed during a printing of the visual  
medium materials.

25

20. The file structure of claim 17, wherein a setting of the both state option  
causes a low resolution graphical element to be embedded in the Print Ready File, with a  
subsequent link to a high resolution graphical file, whereby the low resolution graphical  
element imaged during a preview of the visual medium materials, and the high resolution  
30 graphical file is printed during a printing of the visual medium materials.



21. A method for producing consistent visual medium materials for a customer order, the method comprising:

5     collecting and storing static and variable customer information describing the visual medium materials;

collecting and storing rule information pertaining to a customer order;

utilizing the rule information to compose the customer information into a Print Ready File;

10     storing the Print Ready File for retrieval and use by at least one device for producing a plate file, whereby the plate file may be used to produce the visual medium materials.

22. The method of claim 21, wherein the Print Ready File is a single file.

23. The method of claim 21, wherein the at least one device for producing a plate  
15     file includes a batcher and plater.

24. The method of claim 21, further comprising:

receiving the plate file by a central vendor.

20     25. The method of claim 24, further comprising:

distributing the plate file to subsequent vendors via a transmission medium.

26. The method of claim 25, further comprising:

25     sending the plate file through a raster image processor for subsequent display or printing of the visual medium materials.

27. The method of claim 25, wherein the transmission medium includes the Internet.



28. A system for producing consistent visual materials for a customer, said system comprising:

5 a web server computer arranged to receive a description of said visual materials from a customer;

a database that stores said description of said visual materials;

10 a Print Ready File that includes text and line elements representing said visual materials, said Print Ready File including a global flag indicating an imaging condition for all of said elements, each of said elements including a state flag indicating an imaging condition for each element;

an asset server computer that stores said Print Ready File; and

15 a plater server computer arranged to accept said Print Ready File and to perform imposition, thus producing a plate file, whereby said Print Ready File may be used to both preview and to print said visual materials for said customer.

29. The system of claim 28 wherein said database stores imposition requirements used by said plater server computer to produce said plate file.

20 30. The system of claim 28 wherein said Print Ready File further includes comparison logic used to compare said global flag to said state flags to determine whether each of said elements should be imaged for a preview or a print situation.

31. A method of printing visual materials for a customer comprising:  
25 receiving a description of said visual materials from said customer;  
storing said description in a database;  
receiving imposition requirements from said customer relating to said visual materials;  
storing said imposition requirements in said database;  
imposing a plate file using said description and said imposition requirements;



sending said plate file to a print vendor.

32. The method of claim 31 further comprising;

5 using a Print Ready File to impose said plate file, said Print Ready File including text and line elements representing said visual materials, said Print Ready File including a global flag indicating an imaging condition for all of said elements, and each of said elements including a state flag indicating an imaging condition for each element.



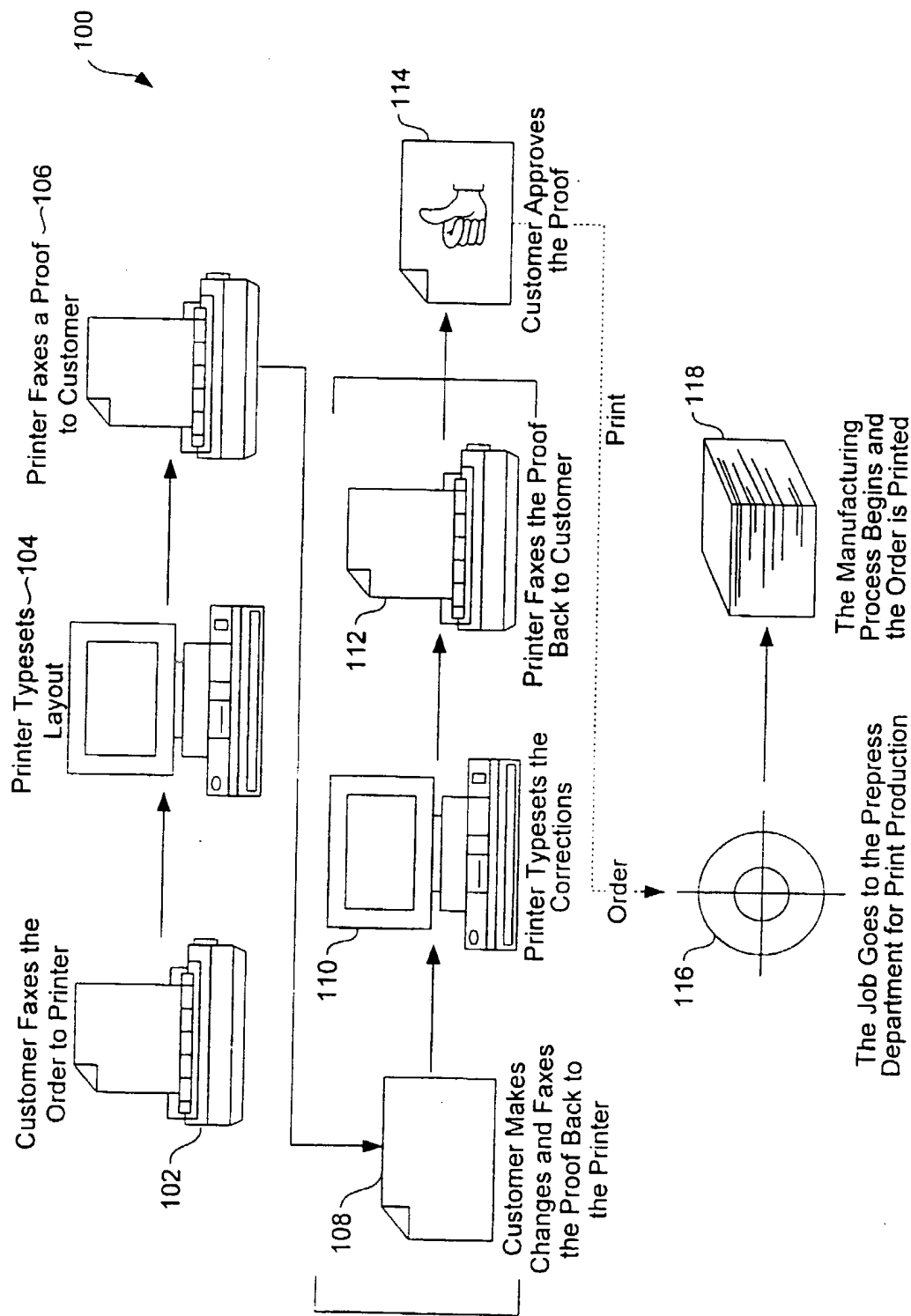


FIG. 1 (Prior Art)



2/21

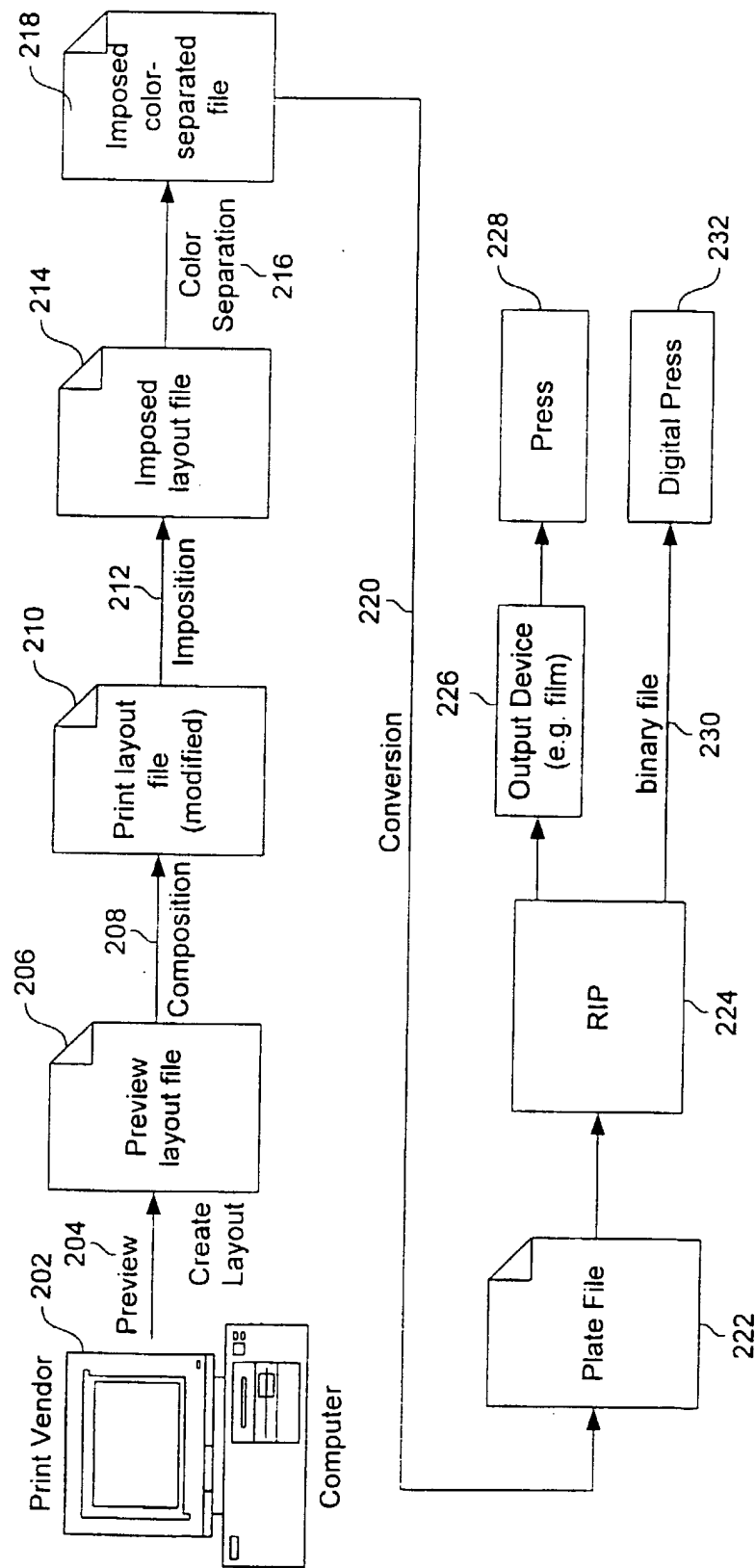


FIG. 2 (Prior Art)



3/21

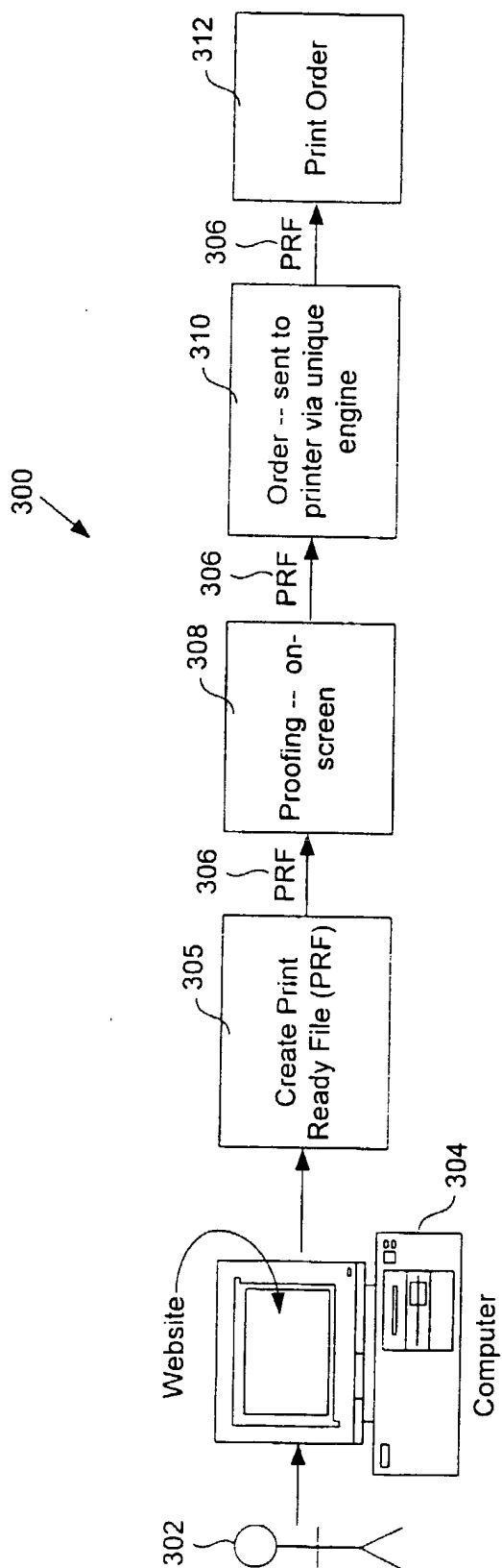
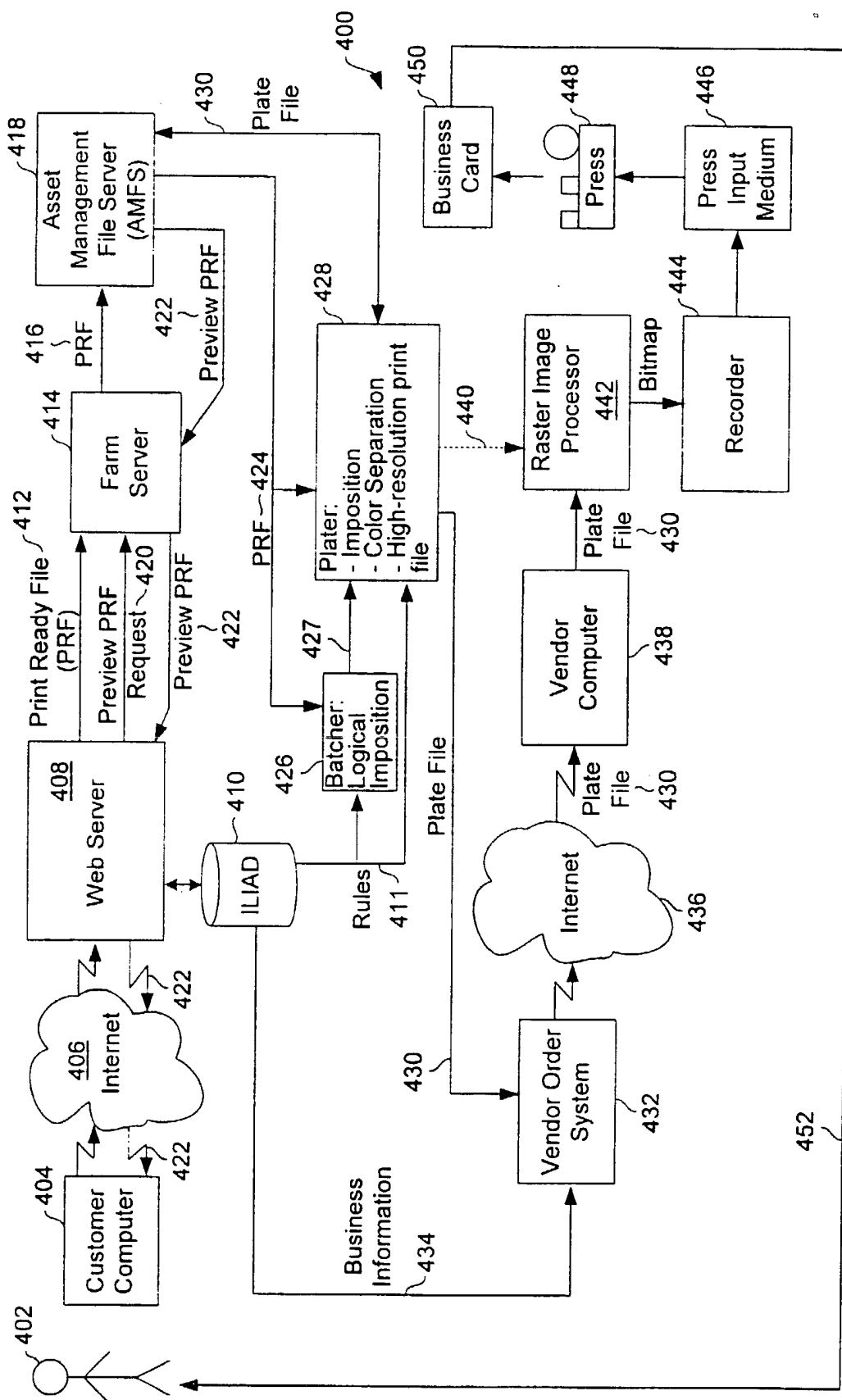


FIG. 3





**FIG. 4**



5/21

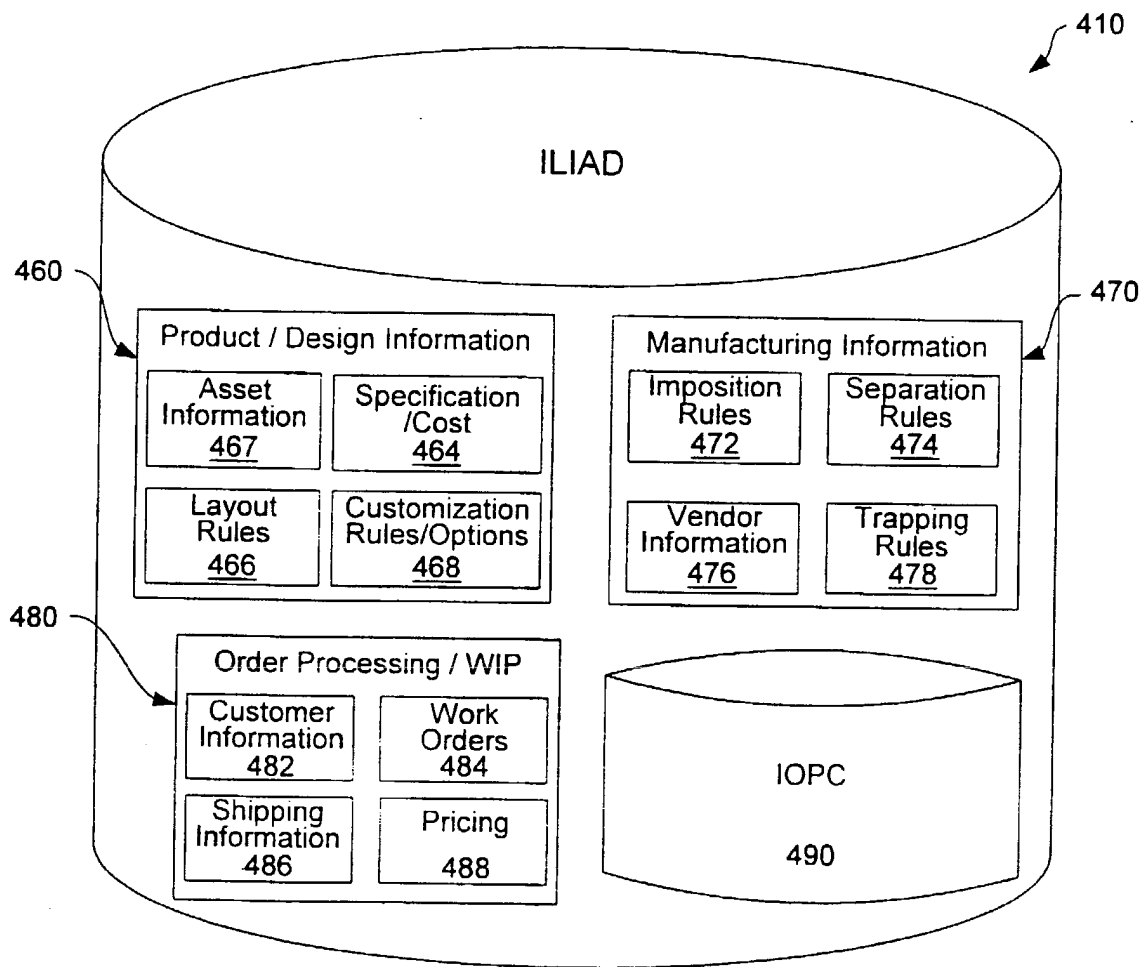


FIG. 4A



6/21

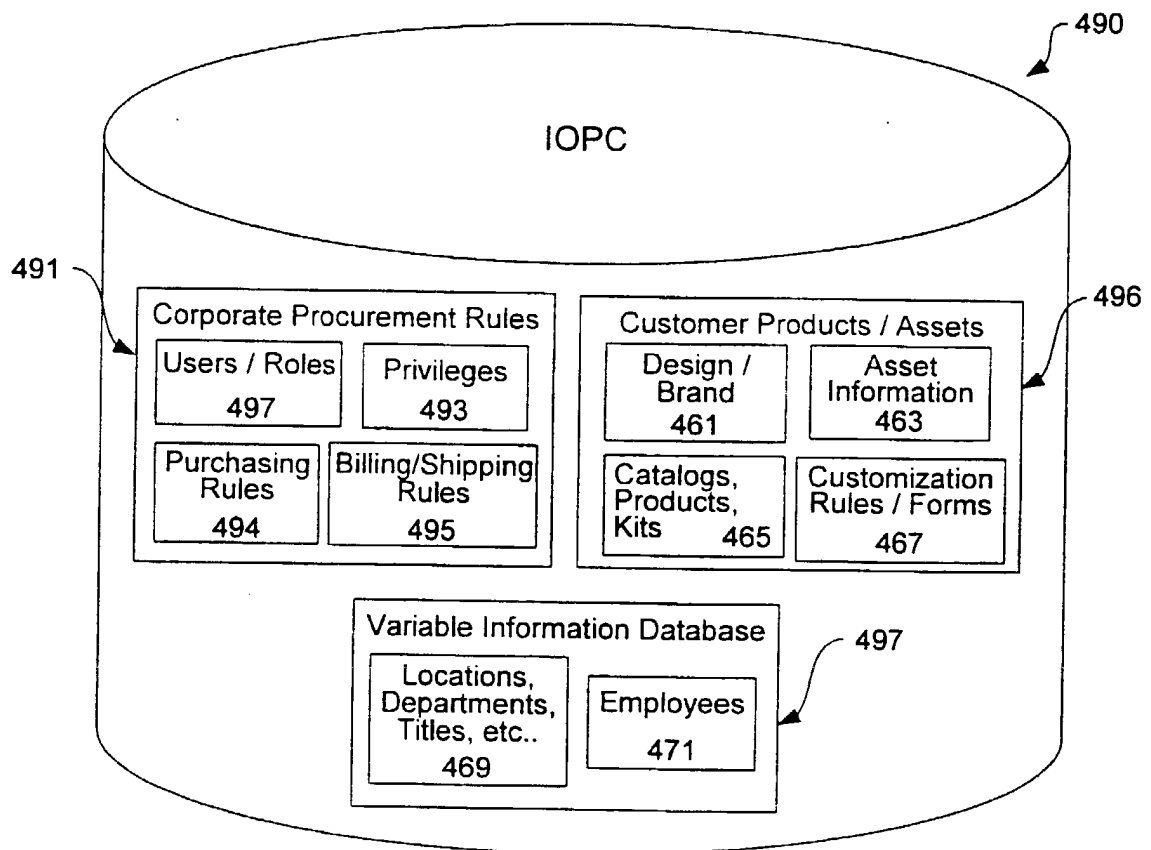


FIG. 4B



7/21

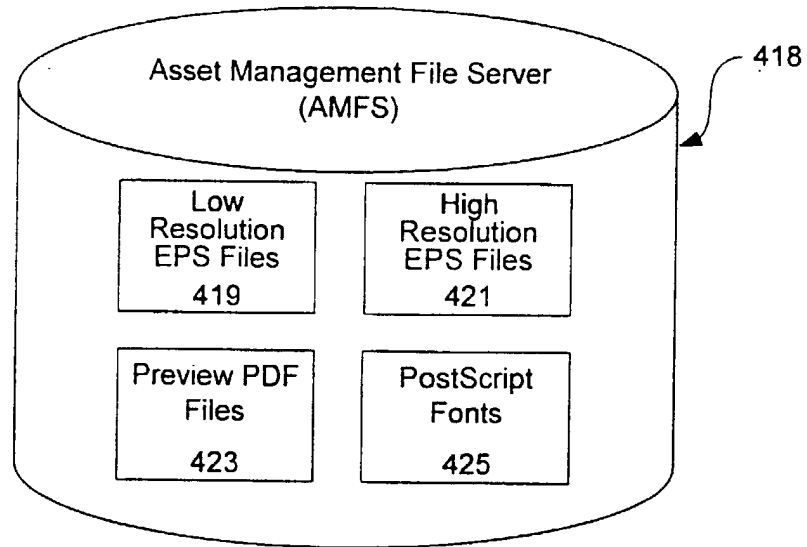


FIG. 4C



8/21

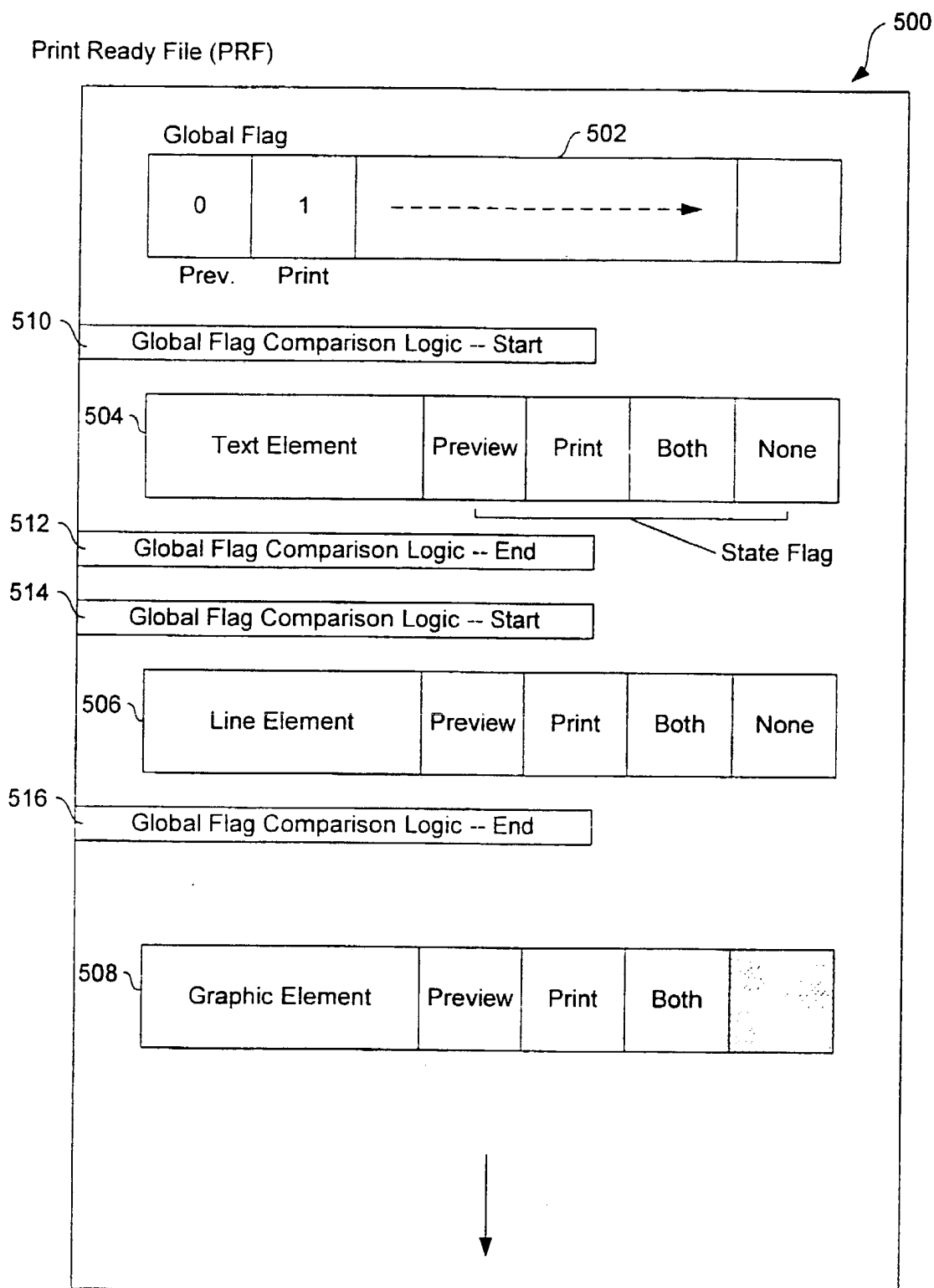


FIG. 5A



9/21

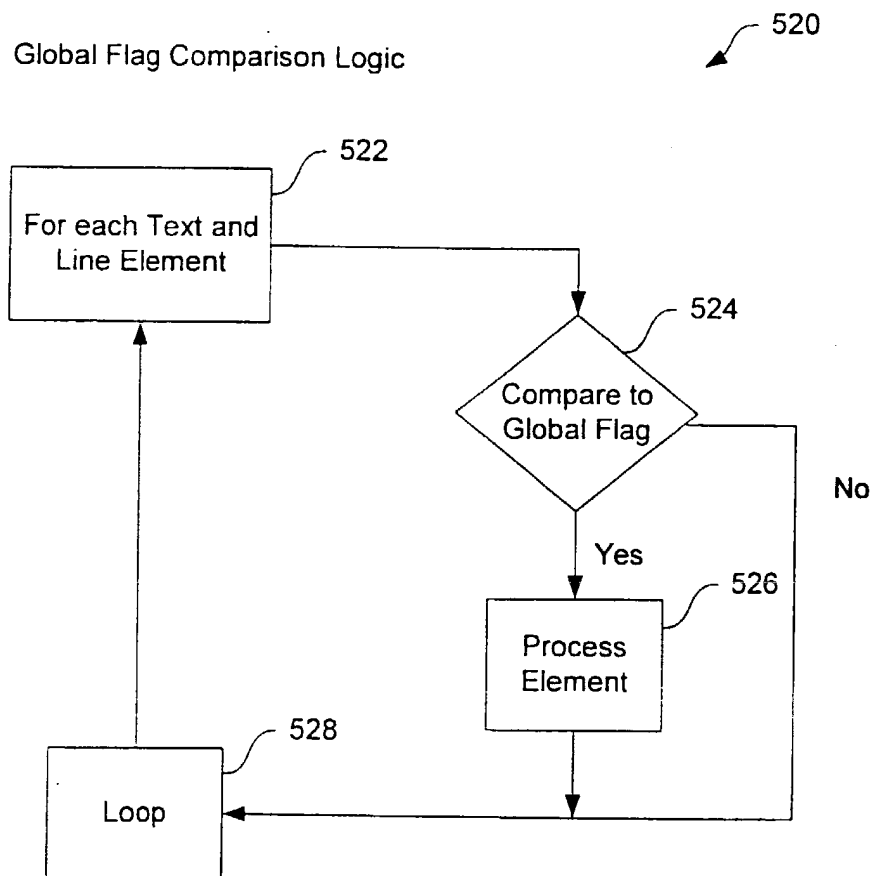


FIG. 5B



10/21

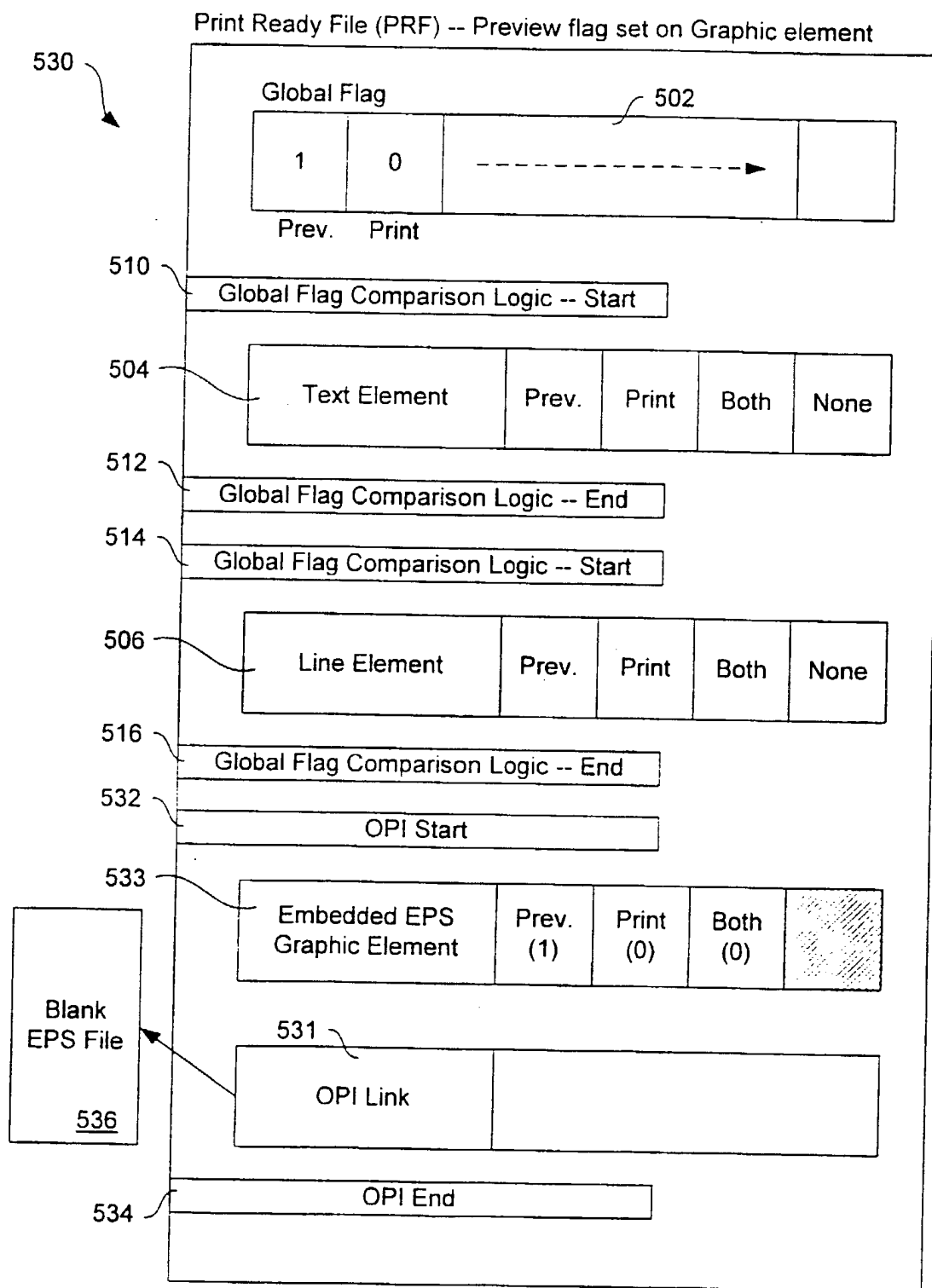


FIG. 5C



11/21

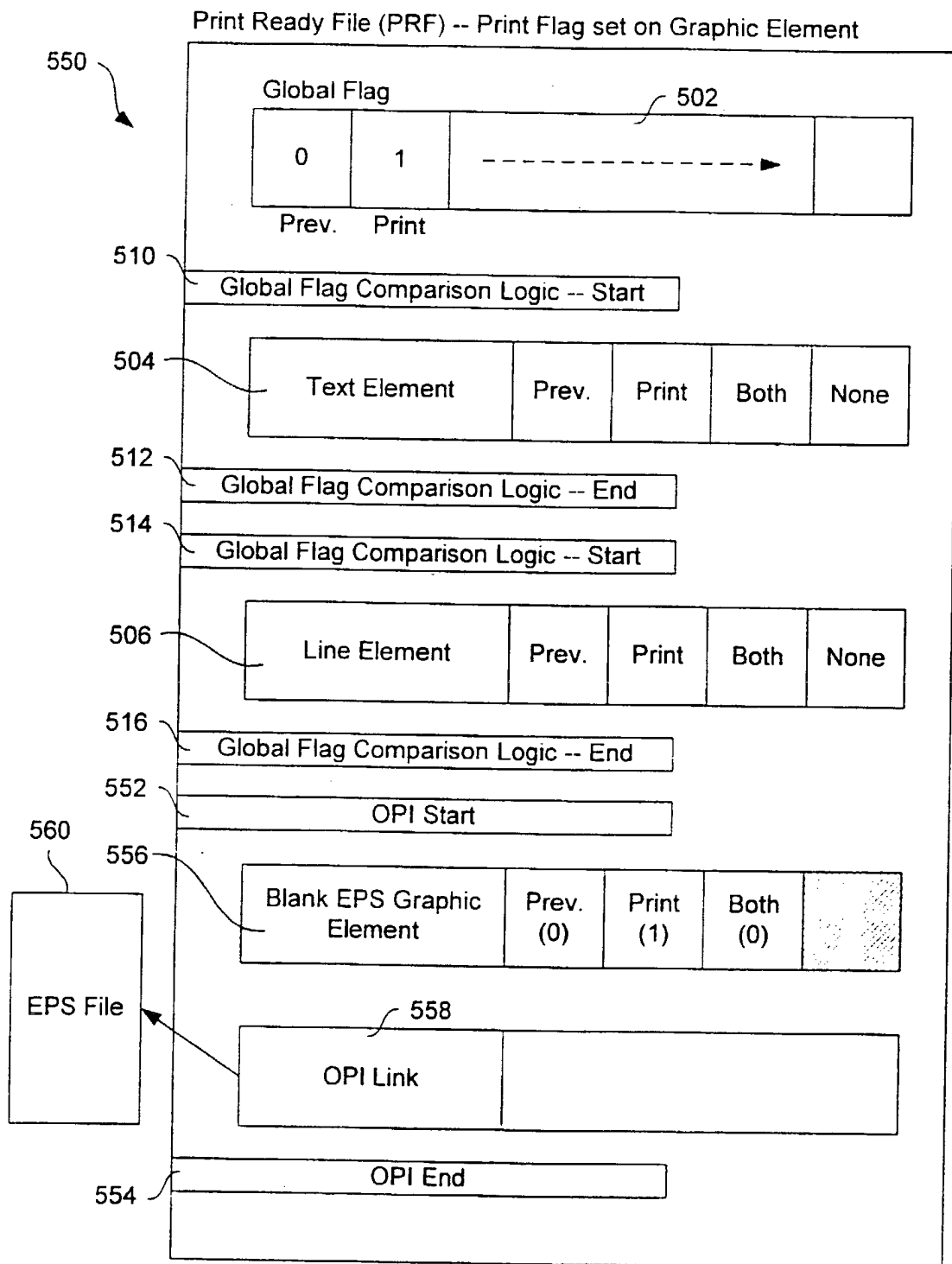


FIG. 5D



12/21

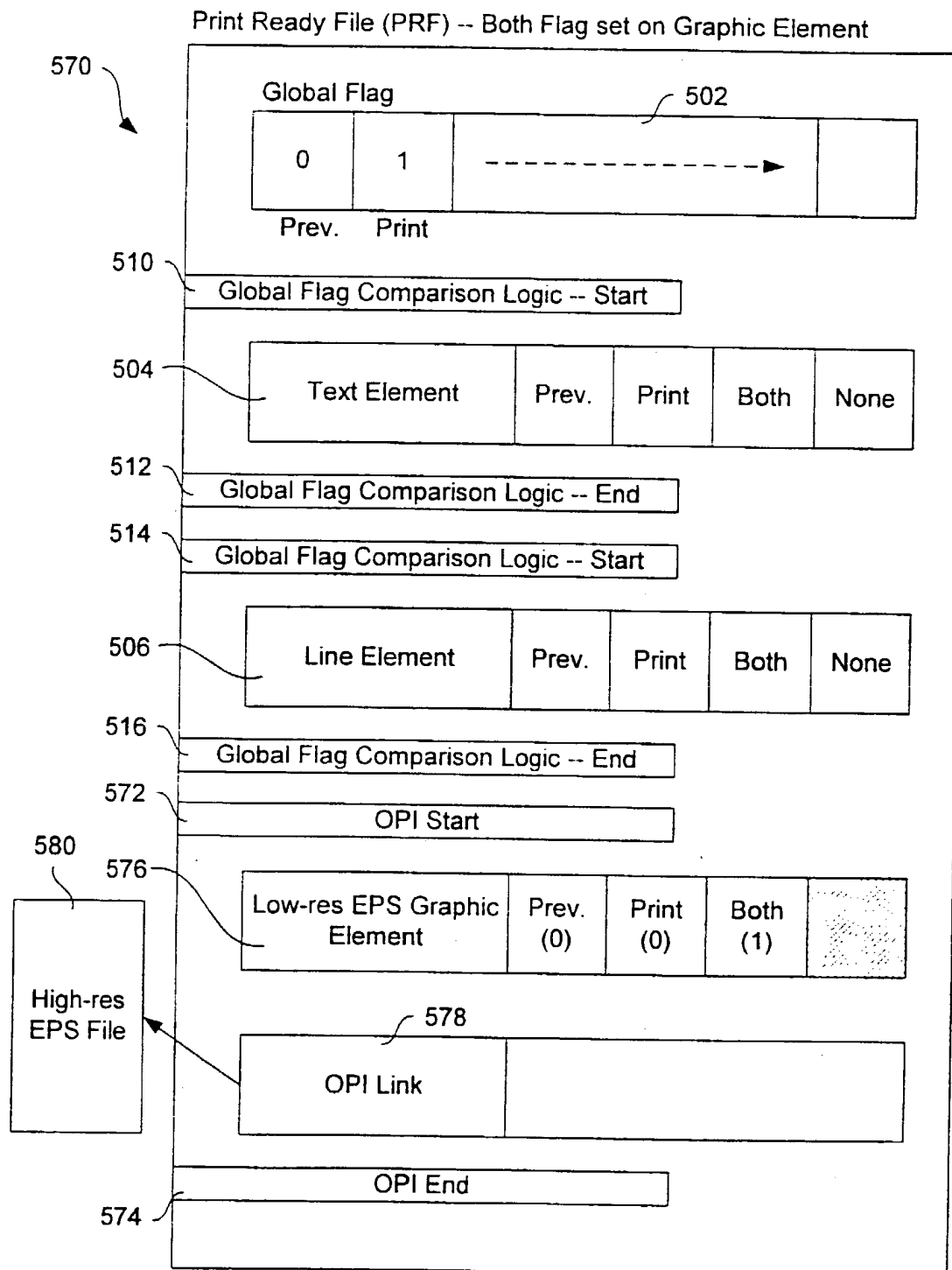


FIG. 5E



13/21

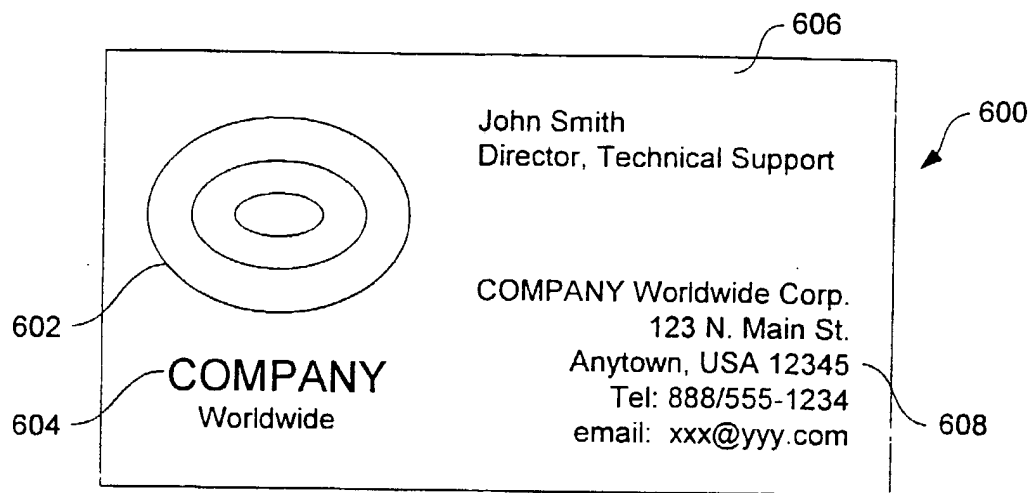


FIG. 6

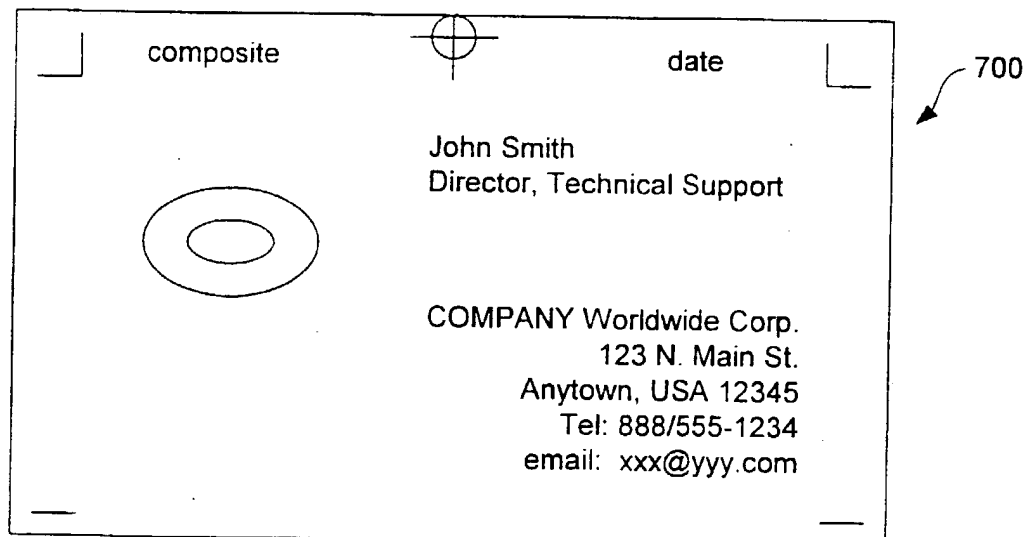
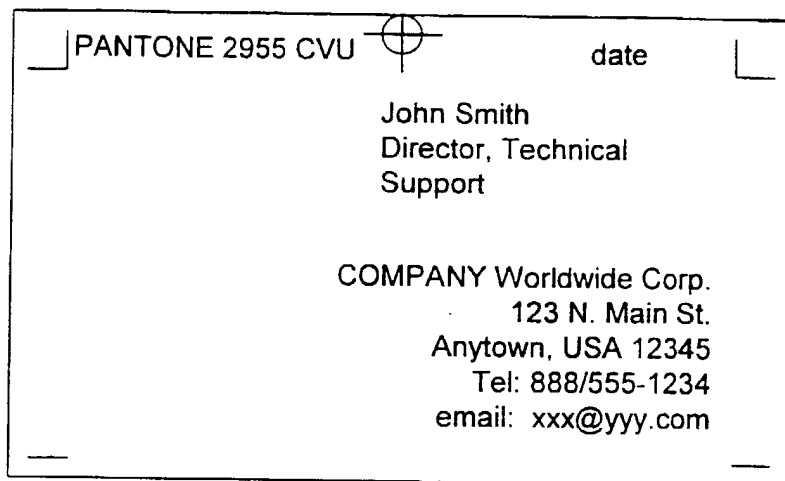


FIG. 7

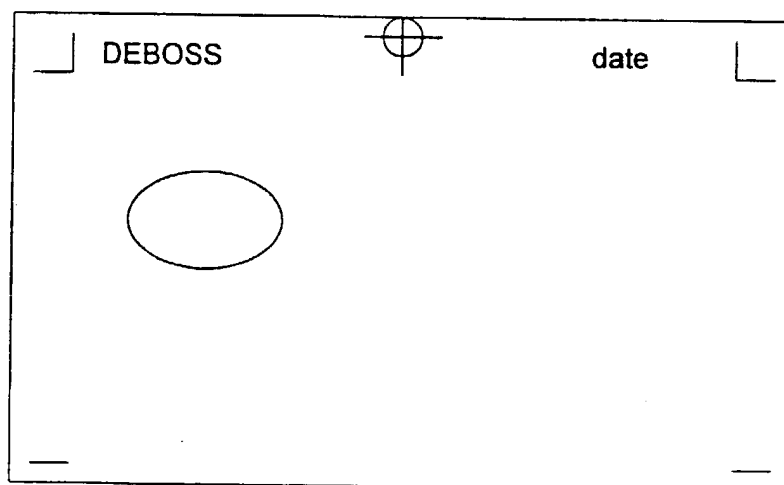


14/21



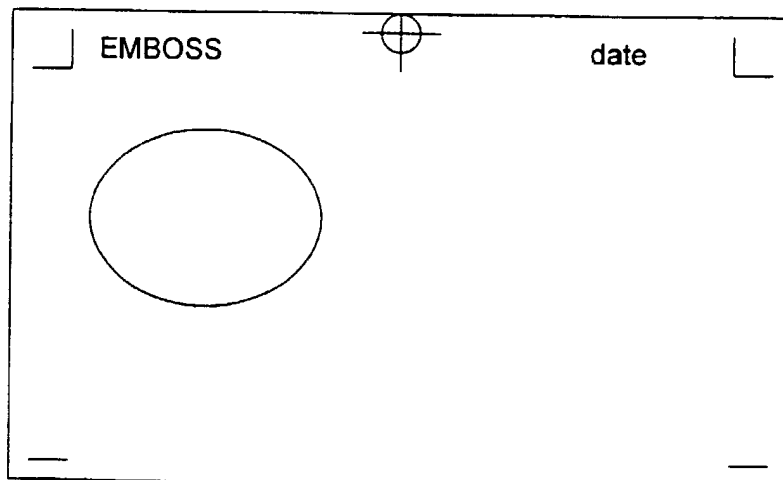
810

FIG. 8A



820

FIG. 8B



830

FIG. 8C

SUBSTITUTE SHEET (RULE 26)



15/21

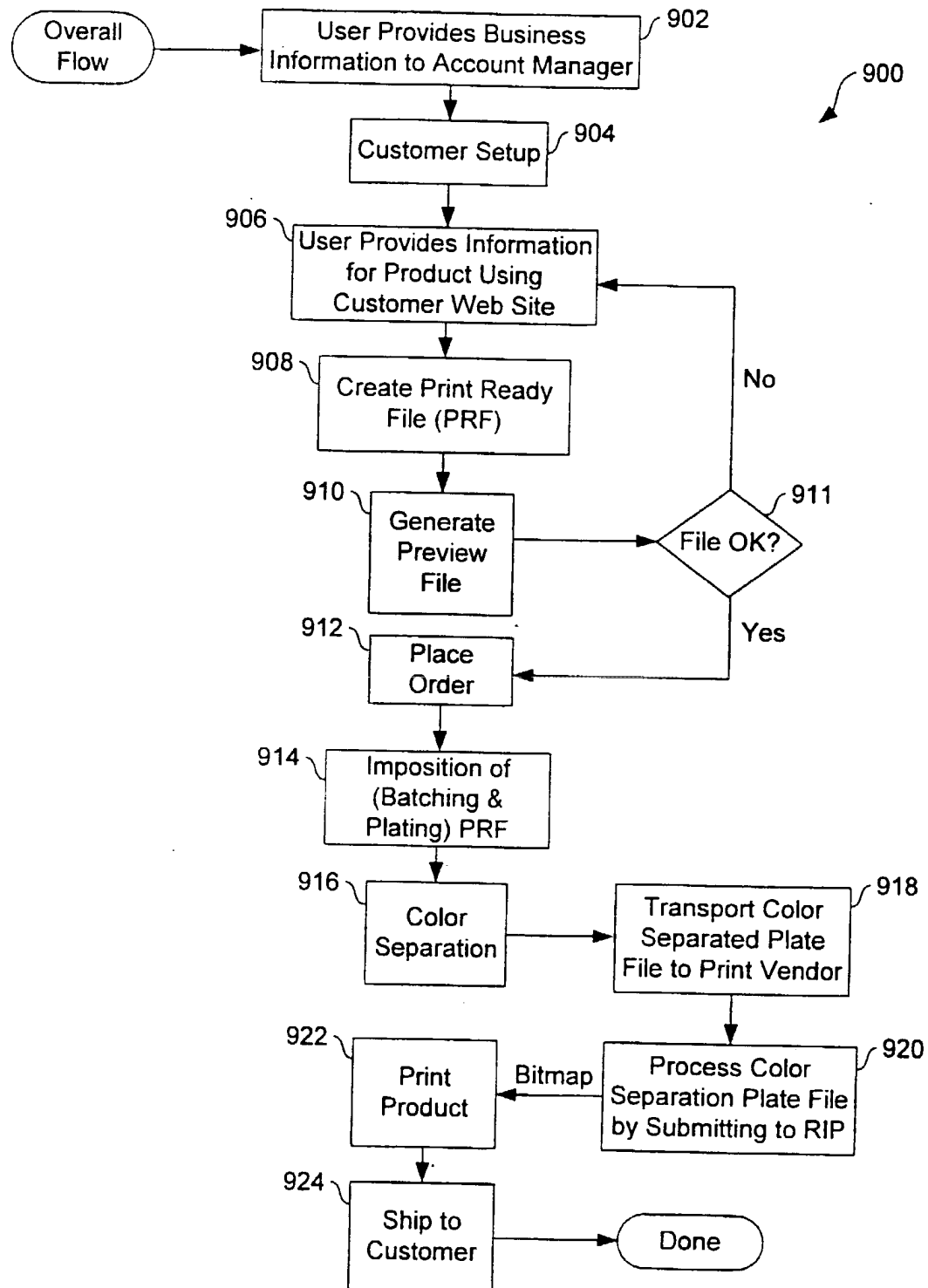


FIG. 9

SUBSTITUTE SHEET (RULE 26)



16/21

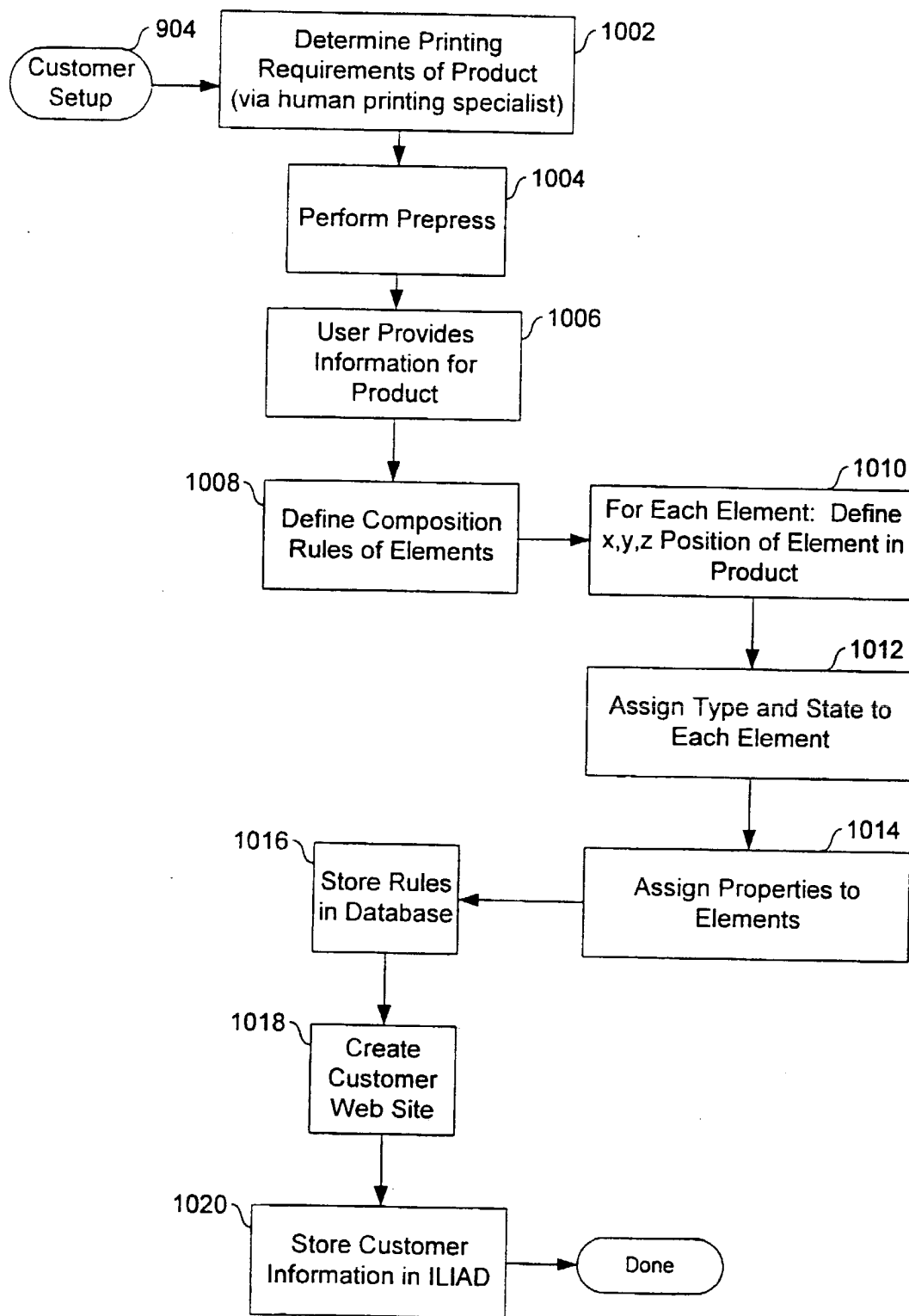


FIG. 10

SUBSTITUTE SHEET (RULE 26)



17/21

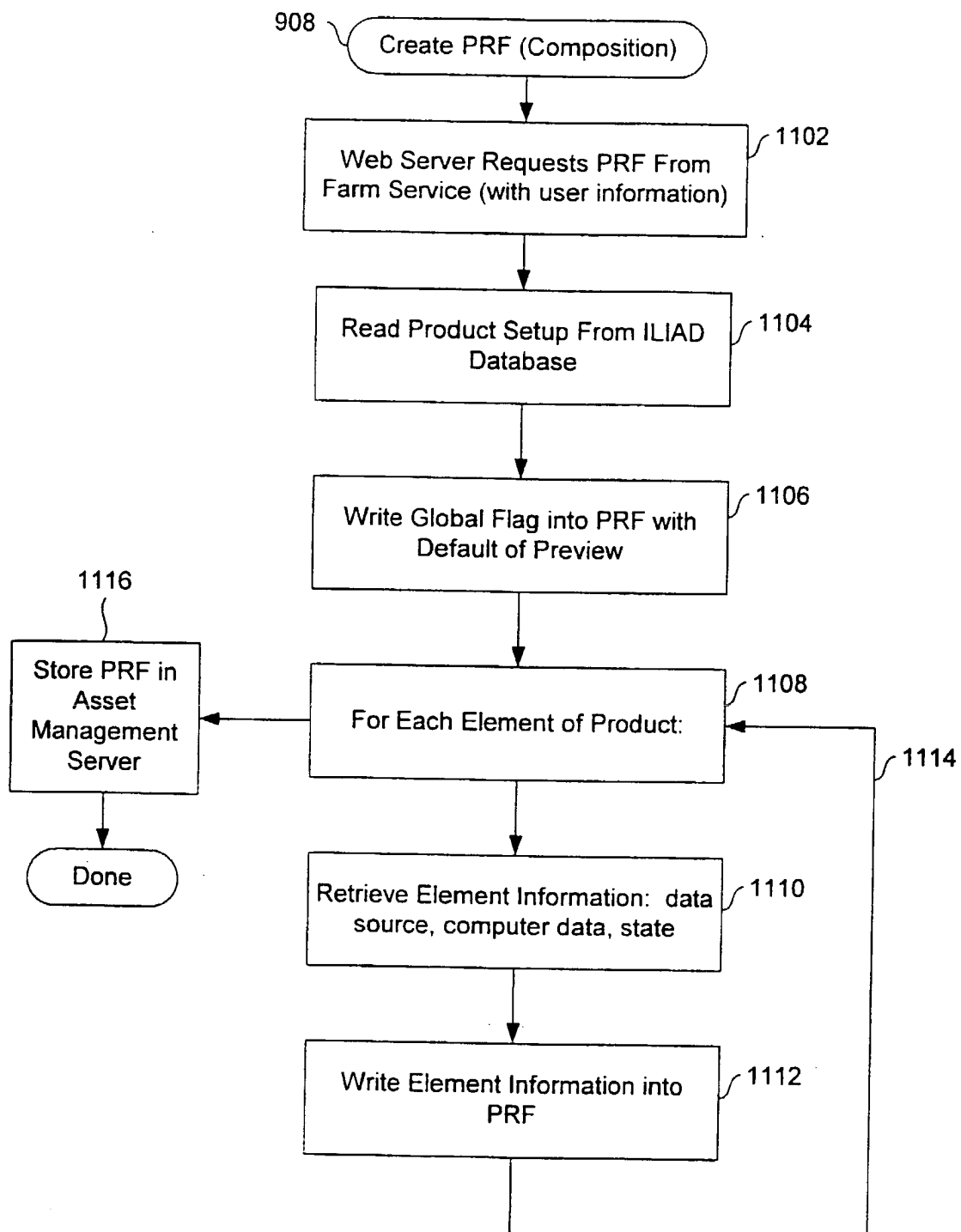


FIG. 11



18/21

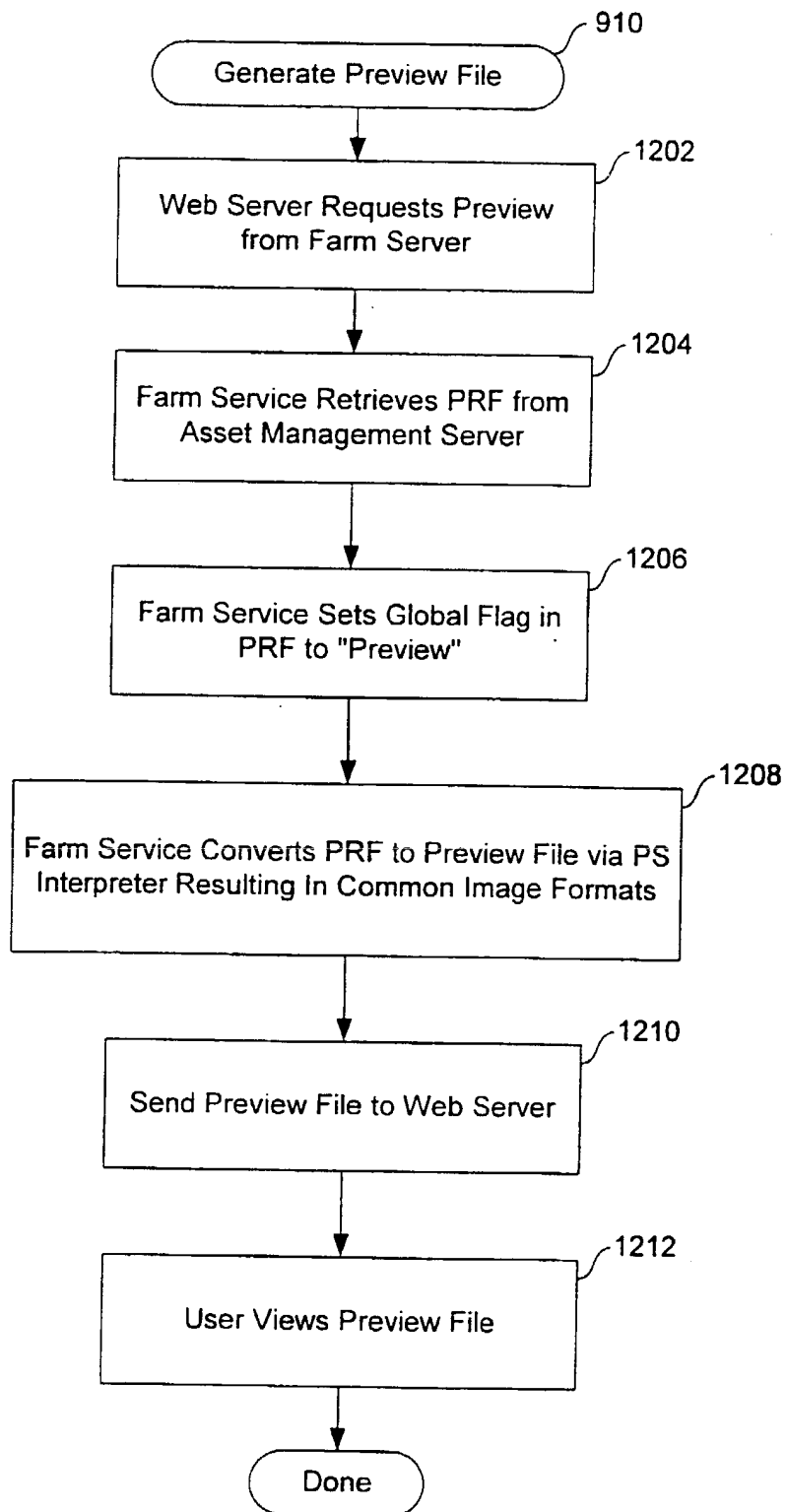


FIG. 12



19/21

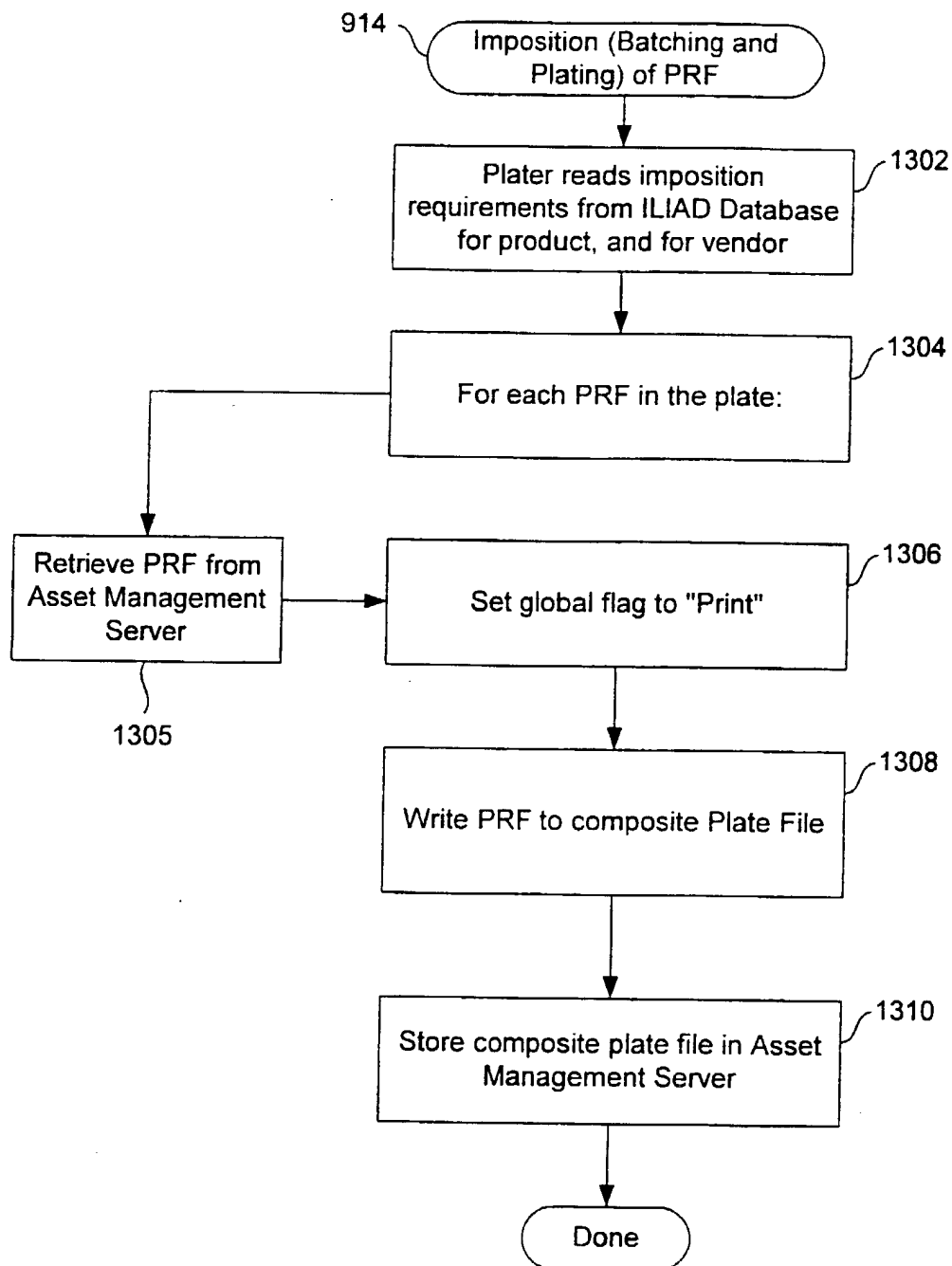


FIG. 13



20/21

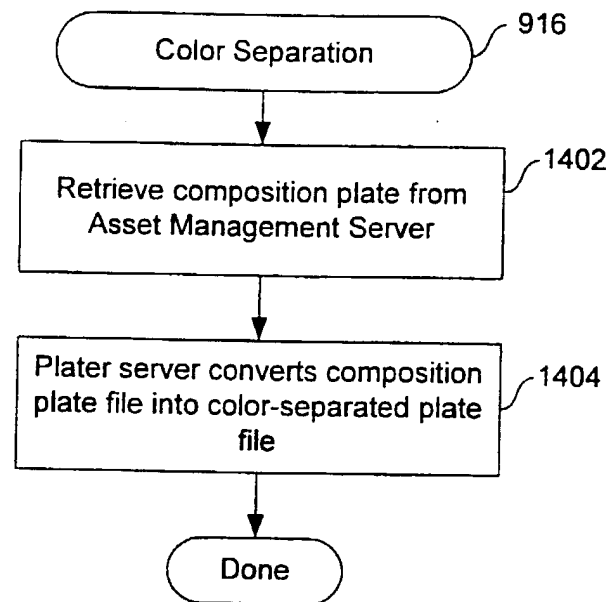


FIG. 14



21/21

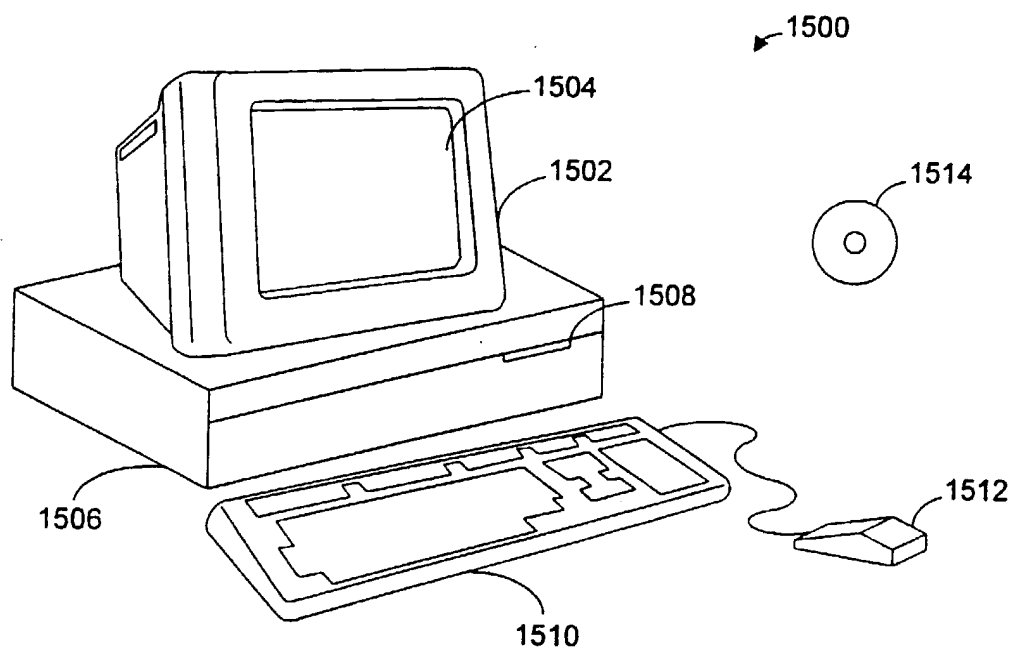


FIG. 15A

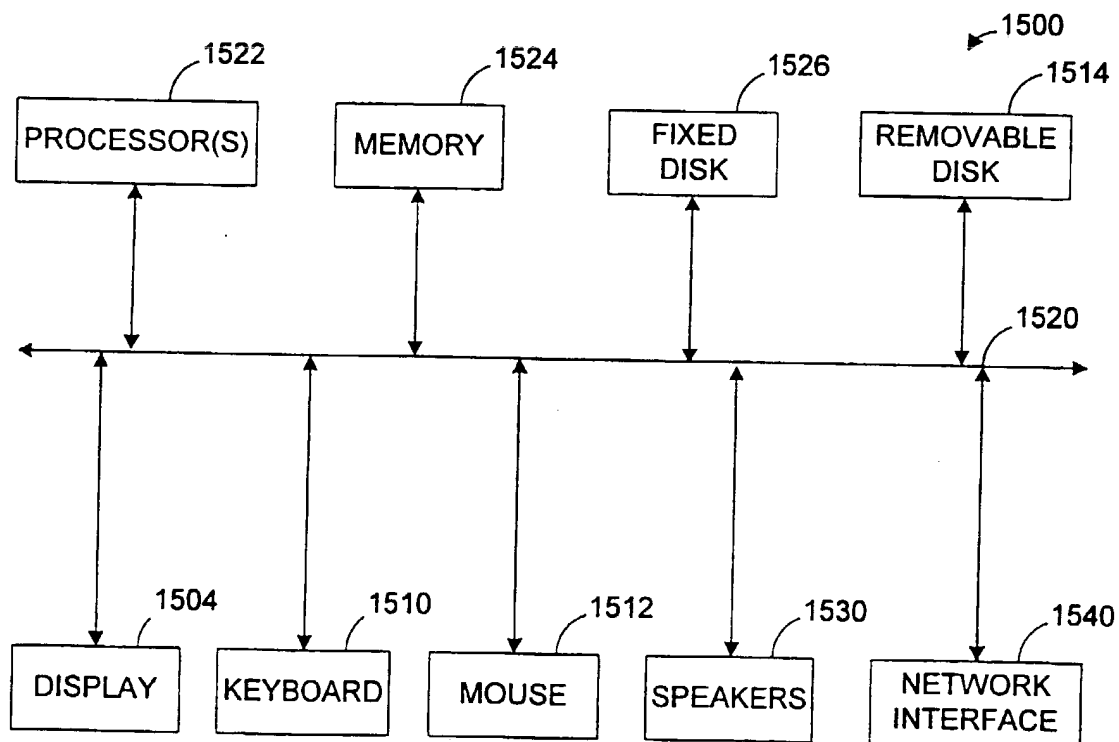


FIG. 15B



# INTERNATIONAL SEARCH REPORT

Inter national Application No

PCT/US 00/10456

## A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G06F17/30 G06F17/60 G06K15/00

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06F G06K

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ, INSPEC, IBM-TDB

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	EP 0 782 068 A (DELUXE CORP) 2 July 1997 (1997-07-02)  abstract page 2, line 58 -page 4, line 57; figures 1,2 page 21, line 24 - line 53; figure 9 page 28, line 25 -page 29, line 39 ---	21-23,31  1-6,27
X A	EP 0 537 391 A (MOORE BUSINESS FORMS INC) 21 April 1993 (1993-04-21)  abstract column 1, line 57 -column 4, line 6 column 5, line 35 -column 7, line 43; figure 1 column 12, line 35 -column 13, line 33; figures 10A,10B,11 ---	21-24,31  25-27
	-/-	

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

### \* Special categories of cited documents :

- "A" document defining the general state of the art which is not considered to be of particular relevance
- "E" earlier document but published on or after the international filing date
- "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- "O" document referring to an oral disclosure, use, exhibition or other means
- "P" document published prior to the international filing date but later than the priority date claimed

- "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- "&" document member of the same patent family

Date of the actual completion of the international search

21 August 2000

Date of mailing of the international search report

28/08/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

Polzer, A



## INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 00/10456

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 837 401 A (SCITEX DIGITAL PRINTING INC) 22 April 1998 (1998-04-22) abstract column 2, line 35 -column 3, line 17 column 4, line 32 -column 7, line 39; figures 1,2 ---	1,2,5, 21,22
X A	WO 98 08176 A (MOORE BUSINESS FORMS INC) 26 February 1998 (1998-02-26)  abstract page 3, line 24 -page 4, line 21 page 5, line 25 -page 9, line 5; figures 1,1A ---	21,22  1,7,17, 28,31,32
X	"BUILD-A-CHECK" IBM TECHNICAL DISCLOSURE BULLETIN,US,IBM CORP. NEW YORK, vol. 34, no. 6, 1 November 1991 (1991-11-01), pages 32-33, XP000228340 ISSN: 0018-8689 the whole document -----	21,22



# INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 00/10456

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0782068 A	02-07-1997	AU 7652496 A CA 2194106 A JP 10031785 A	03-07-1997 30-06-1997 03-02-1998
EP 0537391 A	21-04-1993	AU 662237 B AU 646894 B CA 2052813 A,C NZ 240203 A AU 4494893 A AU 8572191 A NZ 250294 A	24-08-1995 10-03-1994 05-04-1993 26-09-1995 02-12-1993 03-06-1993 28-05-1996
EP 0837401 A	22-04-1998	CA 2215094 A	16-04-1998
WO 9808176 A	26-02-1998	AU 3982097 A BR 9712789 A CN 1231750 A EP 0920667 A NZ 334700 A	06-03-1998 14-12-1999 13-10-1999 09-06-1999 29-07-1999